НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»

УДК 004.056

«До захисту допущено»
В.о. завідувача кафедри
_____ М.В.Грайворонський
“___”_____2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності: 125 Кібербезпека

на тему: Ідентифікація Twitter ботів засобами машинного навчання

Виконав: студент 2 курсу, групи ФБ-71мн
(шифр групи)

Кіфорчук Кирило Олегович _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник доцент кафедри ІБ, к.т.н., Родіонов А. М. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою
Спеціальність (спеціалізація) – 125 Кібербезпека («Системи, технології та математичні методи кібербезпеки»)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«___»_____2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кіфорчуку Кирилу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема дисертації Ідентифікація Twitter ботів засобами машинного навчання ,
науковий керівник дисертації Родіонов Андрій Миколайович, к.т.н. ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «02» квітня 2019 р. № 1023-с

2. Термін подання студентом дисертації 06.05.2019 р.

3. Об'єкт дослідження виявлення ботів в соціальній мережі Twitter

4. Предмет дослідження ключові характеристики облікових записів ботів в соціальній мережі Twitter

5. Перелік завдань, які потрібно розробити
- аналіз існуючих моделей
- формування навчальної вибірки
- знаходження класифікуючих метрик
- вибір оптимального алгоритму навчання
- перевірка якості побудованої моделі

6. Орієнтовний перелік ілюстративного матеріалу 32 ілюстрації

7. Орієнтовний перелік публікацій

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд літературних джерел	01.09.2017 – 10.06.2018	виконано
2	Аналіз існуючих рішень	03.09.2018 – 01.12.2018	виконано
3	Формування навчальної вибірки	01.12.2018 – 01.01.2019	виконано
4	Побудова моделі ідентифікації	01.01.2019 – 01.04.2019	виконано
5	Перевірка якості розробленого рішення	01.04.2019 – 10.05.2019	виконано

Студент _____ Кіфорчук К.О.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Родіонов А.М.
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Робота обсягом 71 сторінка містить 32 ілюстрації, 5 таблиць та 14 літературних джерел.

Актуальність роботи зумовлюється браком точності та якості існуючих рішень з розпізнавання ботів. Аналіз моделей ідентифікації автоматизованих облікових записів мережі Twitter виявив такі недоліки: застарілість та неповнота навчальної вибірки, обмеженість набору характеристичних особливостей, недостатня точність та гнучкість існуючих моделей.

Метою дослідження є створення моделі ідентифікації ботів у соціальній мережі Twitter. Для досягнення цієї мети було поставлено наступні завдання:

1) проаналізувати існуючі моделі ідентифікації ботів у Twitter;
2) сформувати навчальну вибірку для машинного навчання;
3) дослідити закономірності між класом об’єкта та його параметрами;
4) на основі попереднього кроку вибрати параметри, що класифікують обліковий запис;
5) зробити оптимальній вибір алгоритму машинного навчання;
6) перевірити якість побудованої моделі.

Об’єктом дослідження є виявлення ботів в соціальній мережі Twitter.

Предметом дослідження є ключові характеристики облікових записів ботів в соціальній мережі Twitter.

В якості методів дослідження було обрано: опрацювання літературних джерел, аналіз існуючих моделей, класифікація параметрів облікового запису, вимірювання якості та точності обраних метрик, методи машинного навчання, моделювання процесу ідентифікації ботів та обробка отриманих результатів.

Наукова новизна отриманих результатів:

- удосконалено метод формування навчальної вибірки для алгоритмів машинного навчання;

- вперше було враховано характеристичні особливості облікових записів публічних та відомих людей;
- вперше було враховано показники довжини строкових параметрів облікового запису;
- розроблено процес вибору оптимального алгоритму машинного навчання в моделі розпізнавання ботів;
- вперше було побудовано модель ідентифікації ботів з настільки високими показниками точності, повноти та правильності.

Практичне значення отриманих результатів полягає в можливості побудови системи моніторингу активності автоматизованих облікових записів соціальної мережі Twitter. Запропоновані методи та підходи до розробки моделі на основі машинного навчання можуть бути використані для роботи з іншими соціальними мережами.

твіттер, боти, розпізнавання ботів, модель ідентифікації, машинне навчання.

ABSTRACT

The work consists of 71 pages and contains 32 figures, 5 tables and 14 literary references.

The relevance of the work conditioned by the lack of accuracy and quality of existing solutions for identification of Twitter bots. An analysis of identification models of Twitter automated accounts identified the following disadvantages: obsolete and incomplete learning sample, limited set of characteristic features, insufficient accuracy and flexibility of existing models.

The goal of this study is creating a bot identification model in the Twitter. To achieve this goal, the following tasks were set:

1) analyze existing identification models of Twitter bots;
2) form a training sample for machine learning;
3) investigate the patterns between the object class and its parameters;
4) choose the parameters that classify the account based on the previous step;
5) make an optimal choice of machine learning algorithm;
6) check the quality of the built model.

The object of research is detecting Twitter bots.

The subject of research is key features of Twitter bots.

As research methods were chosen: review of literary sources, analysis of existing identification models, classification of account parameters, measurement of quality and accuracy of selected metrics, machine learning methods, modeling of bots identification process and results processing.

The scientific novelty of the results:

- the method of formation of a training sample for algorithms of machine learning was improved;
- for the first time, the characteristics of the accounts of public and well-known people were taken;

- for the first time, the parameters of the length of the account's text features were taken;
- the process of choosing an optimal machine learning algorithm in the bots identification model was developed;
- for the first time, a model for identifying bots with such high levels of accuracy, completeness and accuracy was constructed.

The obtained results can be used for building monitoring system of activity of Twitter automated accounts. The proposed methods and approaches to the development of models based on machine learning can be used to work with other social networks.

twitter, bots, bots identification, identification model, machine learning.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Прикладний програмний інтерфейс (англ. Application Programming Interface)

AUC – Площа обмежена ROC-кривою (англ. Area Under ROC Curve)

ROC-крива – графік, що дозволяє оцінити якість бінарної класифікації (англ. Receiver Operating Characteristic)

ВСТУП

Сучасний світ інформаційної безпеки в багатьох аспектах залежить від людського фактору. Саме людина є однією з найслабших ланок в будь-якій системі захисту. Маніпулювання свідомістю особи або групи осіб - один з найефективніших методів, який використовується в процесі атаки на інформаційну систему.

З ростом популярності соціальних мереж керувати поведінкою людини стало ще простіше, інформаційні війни набули нового масштабу.

Одним з інструментів впливу на громадську думку, а також засобом поширення пропаганди, реклами та фальшивих новин, є автоматизовані облікові засоби у соціальних мережах або боти.

Бот (скорочено від англ. Ro**bot**) – спеціальна програма, яка імітує поведінку реальної особи. Боти в соціальних мережах – це облікові записи, дії в яких виконуються за допомогою програмного забезпечення. Функціональними можливостями таких ботів можуть бути:

- ведення переписки з іншими користувачами;
- наповнення профілю облікового запису інформаційним змістом;
- перегляд змісту облікових записів інших користувачів.

Загалом, автоматизовані облікові записи мають однакові повноваження з реальними користувачами в соціальних мережах та можуть виконувати такі ж самі дії, тому задача розпізнавання ботів не має тривіального вирішення. Проте дії програмного забезпечення в соціальних мережах можуть мати певні характерні особливості. Наприклад, звичайна людина фізично не здатна генерувати десятки повідомлень за секунди, а для програмного забезпечення це є легкою задачею. Швидкість реакції на повідомлення інших користувачів також відрізняється у людини та у бота. Набір кількісних показників подібних характерних особливостей

дозволяє з певною точністю відповісти на питання, керується обліковий запис людиною або програмним забезпеченням.

Однією з найбільших соціальних мереж за кількістю ботів та за загальною кількістю користувачів є Twitter. Цей сервіс було засновано у 2006 році, наразі він налічує близько 1,3 млрд. зареєстрованих та більше 300 млн. активних користувачів. Згідно досліджень, від 9 до 15 відсотків активних користувачів – боти [1]. Характерною особливістю цієї мережі є ведення мікроблогів: кожен користувачів має домашню сторінку, куди він може записувати короткі текстові повідомлення довжиною до 280 символів або розміщувати медіа контент (відео, фотографії, тощо).

В 2016 році діяльність ботів у Twitter посприяла перемозі Дональда Трампа на президентських виборах у США. Після цього багато фахівців з ідентифікації автоматизованих облікових записів зацікавилися задачею розпізнавання ботів саме у соціальній мережі Twitter.

На сьогодні, існує багато онлайн сервісів та програмних бібліотек для розпізнавання Twitter ботів, проте їх спільними недоліками є брак точності визначення та використання великої кількості параметрів облікового запису, що впливає на швидкість обрахування результату.

Зменшення кількості параметрів необхідних для ідентифікації ботів та збільшення точності результату – головні відкриті задачі в сфері розпізнавання автоматизованих облікових записів.

Дана робота присвячена дослідженню та розробці моделі ідентифікації ботів соціальної мережі Twitter, яка б мала високу точність обрахунків при невеликій кількості необхідних параметрів.

1 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ІДЕНТИФІКАЦІЇ БОТІВ

Існує декілька принципових способів представлення програмного забезпечення для ідентифікації ботів у Twitter: онлайн-сервіс та програмна бібліотека. В цьому розділі розглядаються обидва способи представлення моделі розпізнавання ботів, а також надаються короткі відомості про теоретичне підґрунтя описаних моделей.

1.1 Методика машинного навчання

Всі моделі ідентифікації ботів у Twitter розглянуті в цьому розділі використовують інструменти машинного навчання. Нижче викладено короткі теоретичні відомості стосовно використання та основної ідеї цих методів.

Загальна постановка задачі машинного навчання: є деяка множина об’єктів (ситуацій) та множина можливих відповідей на ці ситуації, потрібно встановити закономірність між вхідними параметрами та кінцевими відповідями.

Залежно від наявності точно встановлених прикладів «параметри-відповідь» методи машинного навчання поділяються на дві широкі категорії: навчання з учителем та без. Такі приклади називають тренувальною або навчальною вибіркою. Моделі представлені в даному розділі ґрунтуються на машинному навчанні з учителем, тобто в їх основі лежить деякий набір даних з характеристиками облікового запису та відома кінцева відповідь чи є користувач ботом. Використовуючи ці вибірку для «навчання» обчислювальної одиниці за певним алгоритмом, можна дати таку ж кінцеву відповідь для іншого набору даних, який не має мітки «бот» або «людина».

Алгоритми машинного навчання з учителем детально розглянуто в розділі 2 даної роботи.

1.2 Онлайн сервіс Botcheck.me

Труднощі ідентифікації ботів на платформі соціальних медіа, таких як Twitter, постають у тому, що не існує детермінованого способу дізнатися, як виглядає бот, тобто визначити повний та вичерпний набір параметрів, яким ботів можна точно описати. На відміну від академічних наборів даних, для цих облікових записів не існує достовірних знань, міток або підстав, за допомогою яких можна точно визначити, чи керується обліковий запис автоматизованим програмним забезпеченням. Це проблема курки та яйця: не можна ідентифікувати ботів, якщо ви не знаєте, як виглядають боти, і ви не знаєте, як виглядають боти, якщо ви не можете визначити ботів [2].

Для того, щоб вирішити цю проблему автори сервісу Botcheck.me, фахівці з компанії RoBhat Labs, ідентифікували облікові записи Twitter з певною підозрілою поведінкою як ботів з високим рівнем довіри (англ. high-confidence bot accounts) [2]. Підозрілою поведінкою, на думку RoBhat Labs, є, наприклад, читання твітів кожні кілька хвилин протягом цілого дня, підтримка поляризаційної політичної пропаганди, поширення фальшивих новин, отримання великої кількості послідовників (англ. followers) за відносно невеликий проміжок часу, а також постійне перепоширення (англ. retweeting) та/або популяризація облікових записів інших ботів з високим рівнем довіри.

Наведені вище евристичні правила не завжди можуть правильно ідентифікувати бота з високим рівнем довіри. Наприклад, знаменитість, яка нещодавно створила обліковий запис, за короткий проміжок часу отримає велику кількість послідовників, а завзятий фанат баскетболу під час Фіналу НБА може робити записи кожні декілька хвилин. Через такі крайові випадки, при розробці сервісу Botcheck.me авторами було використано методи машинного навчання.

Навчальний набір формувався наступним чином: спершу до набору додали класифіковані вручну облікові записи ботів, потім туди увійшли усі послідовники

цих облікових записів, які задовольняли наведеним евристичним правилам. За допомогою такого підходу було згенеровано велику кількість прикладів облікових записів, що є ботами, але при цьому постраждала точність моделі, адже послідовники ботів перевірялися лише декількома простими евристичними правилами.

Для побудови якісної моделі розпізнавання потрібні також облікові записи реальних людей. Такий набір даних автори сервісу отримали за допомогою мітки верифікації від Twitter: отримати маркування «верифікований обліковий запис» можна лише підтвердивши свою особу, тому такі облікові записи з високою точністю можна вважати реальними користувачами.

Окрім задачі ідентифікації ботів загалом, фахівці з RoBhat Labs також окремо виділяють політичних ботів. Саме цей різновид автоматизованих облікових записів мав вплив на вибори президента США в 2016 році.

Для ідентифікації таких ботів використовується значно більше параметрів, ось перелік декількох таких параметрів:

- дата створення облікового запису: спостерігається тенденція до збільшення створених ботів під час інавгурації, при цьому решту часу виборчої кампанії кількість ботів та справжніх користувачів мають однакові тенденції (рисунок 1.1) [3];
- популяризація облікових записів інших ботів;
- кількість послідовників;
- частота написання повідомлень (рисунок 1.2) [3];
- двостороння політична направленість повідомлень: авторами було зясовано, що багато політичних ботів схильні до підтримки поляризаційних політичних сил (рисунки 1.3 та 1.4) [3];
- поширення фальшивих новин;
- зміна імені облікового запису;
- частотні характеристики написання повідомлень.

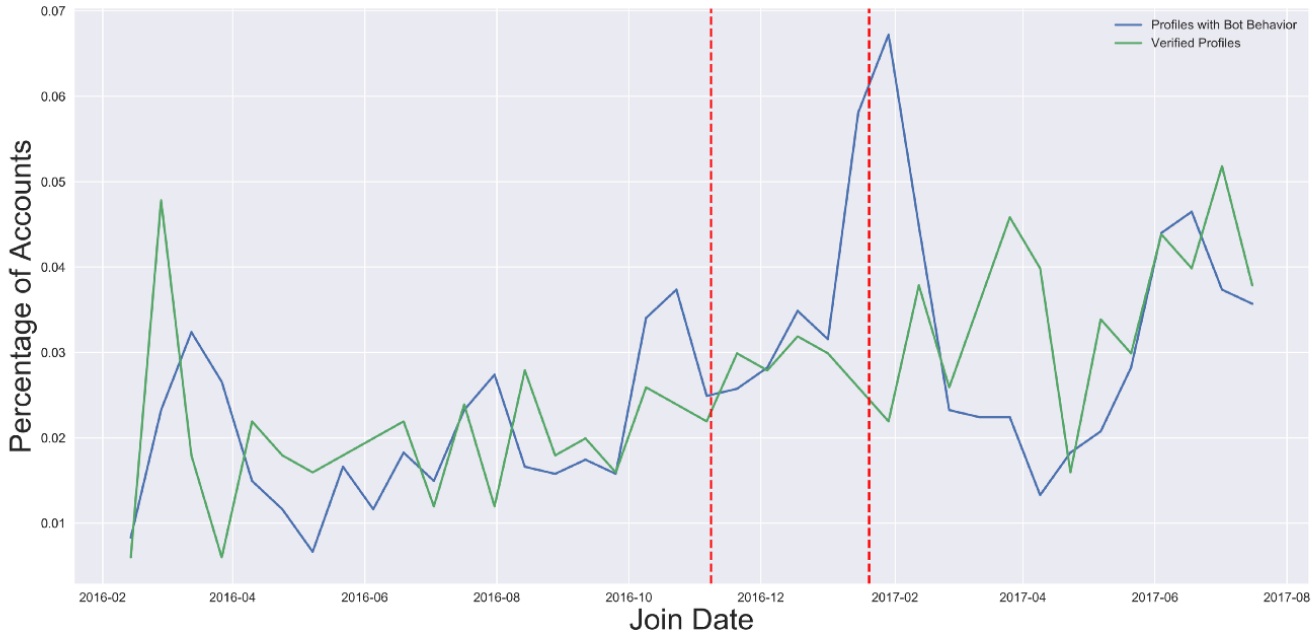

Рисунок 1.1. – Залежність кількості створених облікових записів ботів та перевірених користувачів від дати

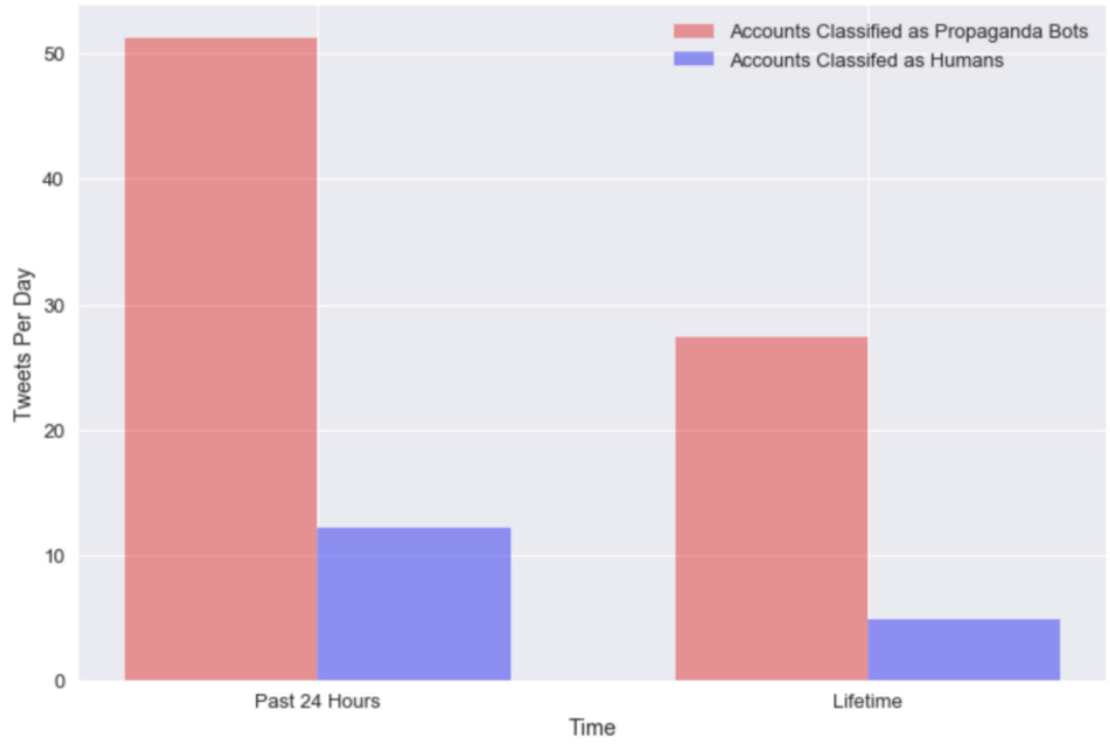

Рисунок 1.2 – Частота повідомлень ботів та перевірених користувачів (останні 24 години та весь час існування облікового запису)

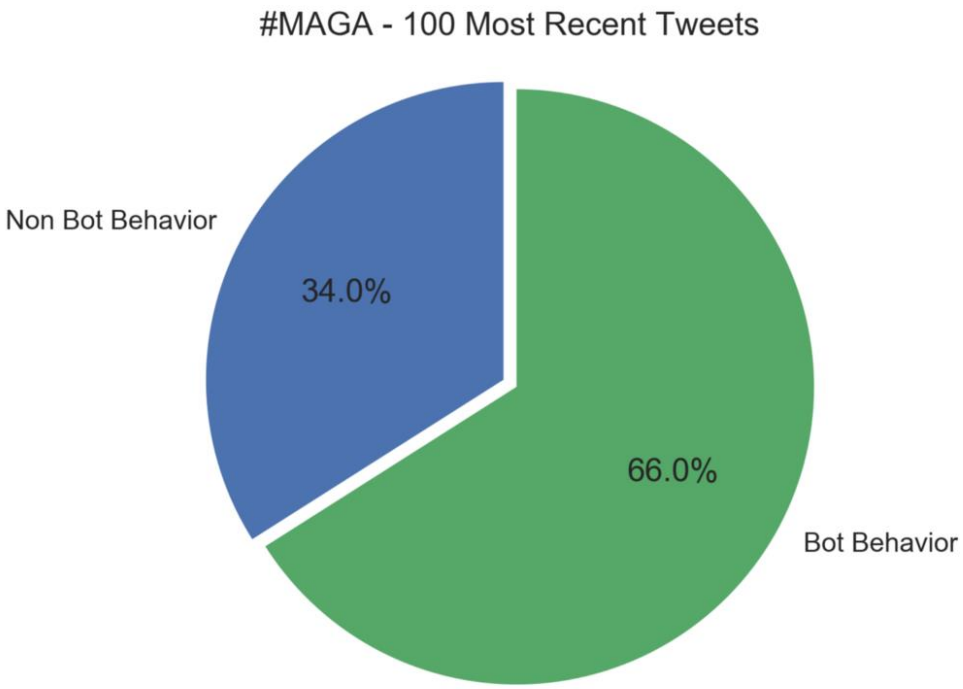

Рисунок 1.3 – Частка ботів серед авторів останніх 100 повідомлень з хештегом #MAGA

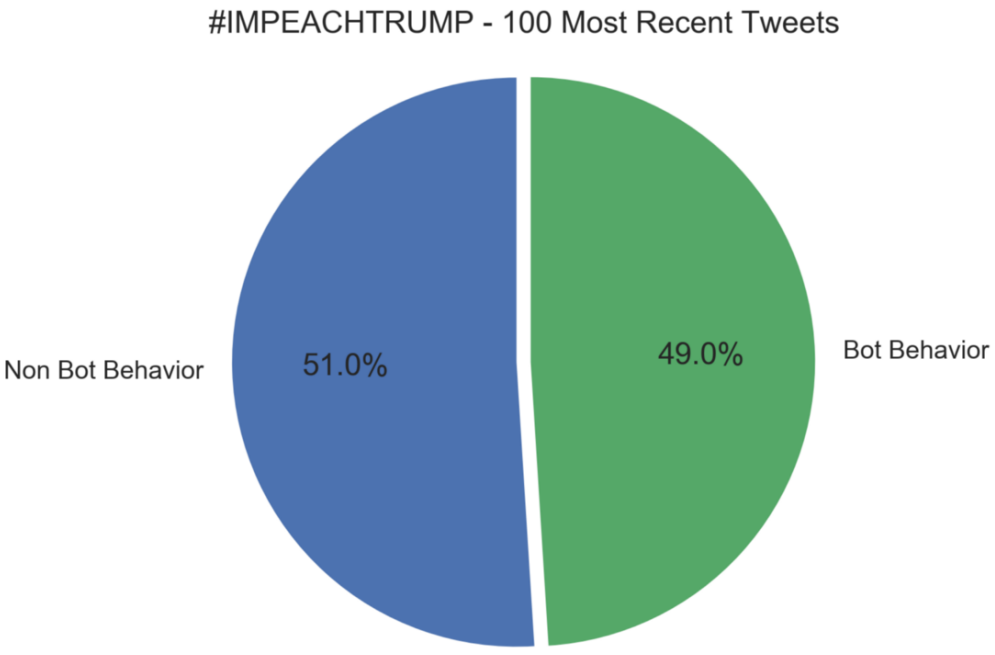

Рисунок 1.4 – Частка ботів серед авторів останніх 100 повідомлень з хештегом #IMPEACHTRUMP

Робота моделі розпізнавання ботів сервісу Botcheck.me перевірялась на облікових записах, що перед тим були перевірені вручну. Точність моделі на цьому наборі становила 0.935. Це ймовірність, з якою автоматизований обліковий запис буде класифіковано як бота.

Впродовж 2 тижнів після запуску сервісу у відкритий доступ було перевірено близько 20 тисяч облікових записів. При цьому похибка першого роду (англ. false positive) становила близько 2%, але, як зазначають самі автори моделі, вона має накопичувальний характер (при збільшенні кількості перевірених облікових записів очікується ріст помилки). Точність визначення ботів при онлайн режимі роботи сервісу становила 0.945.

1.3 Онлайн сервіс Botometer

Інший популярний онлайн-сервіс розпізнавання ботів – система Botometer, що була представлена вченими Університету Індіани та Університету Northwestern.

Вчені запропонували фреймворк, який збирає набір з понад тисячі параметрів та характеристичних особливостей (англ. features) облікового запису користувача Twitter для подальшого використання методів машинного навчання з учителем. Кінцевою відповіддю є число, що належить проміжку [0, 1] та представляє собою ймовірність того, що обліковий запис є автоматизованим.

Авторами моделі було виділено 1150 метрик, які за своєю природою було розділено на шість класів [1]:

- Метрики користувача (англ. User-based features). В цей клас входять особливості, що пов’язані безпосередньо з обліковим записом та його налаштуваннями такі, як геолокація, час створення, кількість друзів, кількість послідовників, опис профілю користувача, кількість повідомлень, налаштування мови, зображення, фонового рисунка, кількість цифр в довжині імені, довжина

імені, кількість відповідей на повідомлення, кількість перепоширень повідомлень інших користувачів, кількість разів, коли перепоширили повідомлення самого користувача, тощо.

- Метрики друзів (англ. Friends features). Соціальна мережа Twitter активно сприяє взаємозв'язку – користувачі пов'язані між собою відношенням «послідовник-друг». Інформаційний вміст подорожує підпорядковуючись саме такому зв'язку, при цьому існує чотири типи посилань до іншого користувача: retweeting, being retweeted, mentioning, being mentioned. Автори моделі згрупували користувачів окремо за кожним видом посилання та для кожної з цих груп зібрали такі метрики, як мова повідомлення, місцевий час користувача, популярність повідомлення. Також в цей клас авторами було включено статистичні величини щодо соціальних контактів користувача (розподіли кількості друзів та послідовників): мінімальне та максимальне значення, медіана, середнє значення, ентропія, стандартне відхилення, коефіцієнт асиметрії, коефіцієнт ексцесу.
- Метрики мережі (англ. Network features). Автори виділили користувачів в окремі соціальні мережі за способом передачі інформації: за допомогою перепоширень, згадувань інших користувачів (англ. mentions) або за допомогою використання хештегів в повідомленнях. Для кожної з виділених мереж було виділено наступні метрики: кількість вершин та ребер, щільність, коефіцієнт кластеризації, метрики централізації, кількість входжень та вихідів з вершини.
- Тимчасові метрики (англ. Temporal features). Фіксують часові шаблони генерації та споживання контенту. Ці параметри включають час між двома послідовними повідомленнями, між двома послідовники згадуваннями інших користувачів та, двома послідовними перепоширеннями повідомлення та статистичні величини розподілів цих метрик (мінімальне та максимальне значення, ентропія, стандартне відхилення, середнє значення, медіана, коефіцієнти асиметрії та ексцесу).

- Метрики контенту та мови (англ. Content and language features). Ці параметри відображають характеристику змісту та мовних особливостей повідомлень користувача. Автори застосували методи розмічування частин мови (англ. POS tagging) для виділення девяти частин мови: дієслова, іменники, прикметники, модальні дієслова, вигуки, прислівники, займенники, запитання (в англійській мові окремо виділяють «wh- words») та пре-детермінатори. Для розподілів частин мови та кількості слів у повідомленні було обраховано статистичні характеристик: мінімальне та максимальне значення, ентропія, стандартне відхилення, коефіцієнти асиметрії та ексцесу, середнє значення та медіана.
- Метрики настроїв (англ. Sentiment features). За допомогою алгоритмів загального призначення оцінки емоцій в тексті авторами було виділено оцінки щастя, збудження, домінування та валентності повідомлень, а також наступні параметри: кількість позитивних та негативних емоцій в одному повідомленні, загальна кількість емоцій в одному повідомленні, відношення емоційних повідомлень до нейтральних, оцінка поляризації одного повідомлення. Також були обраховані оцінки щастя, збудження, домінування та валентності всіх повідомлень користувача та для кожної емоції агрегованих повідомлень обчислено середнє та стандартне відхилення середньозваженого всіх слів лексикону користувача. Для метрик одного повідомлення також обраховано статистичні показники розподілів параметрів: мінімальне та максимальне значення, середнє, медіана, ентропія, стандартне відхилення, коефіцієнти асиметрії та ексцесу.

Для формування навчальної вибірки автори використали 15 тисяч облікових записів ботів з досліджень [4] зібраних за допомогою приманок (англ. honeypots) та 16 тисяч облікових записів реальних осіб. Використавши API Twitter авторами було зібрано до 200 найновіших повідомлень користувача та до 100 найновіших повідомлень інших користувачів, в яких було згадано користувача. Таким чином,

вибірка повідомлень становила близько 3 млн для перевірених користувачів та близько 2,6 млн. для ботів.

Тренувалася модель за допомогою алгоритмів машинного навчання з популярної бібліотеки scikit-learn для мови програмування Python. Точність моделі вимірювалася авторами шляхом обчислення AUC для ROC-кривої з пятикратною перехресною перевіркою (англ. 5-fold cross validation). Були перевірені алгоритми випадкового лісу (англ. Random forest), адаптивного підсилення (англ. Adaptive Boosting), логістичної регресії (англ. Logistic Regression) та дерева ухвалення рішень (англ. Decision Tree). Найкращу точність класифікації показав алгоритм випадкового лісу – точність становила близько 0.95 AUC, для подальших досліджень автори моделі використовували тільки цей алгоритм.

Як зазначають самі автори, точність, що досягалася моделлю на тестовому наборі даних є перебільшеною, оскільки в якості навчальних прикладів використовувалися застарілі облікові записи та, відповідно, застарілі метрики. Щоб перевірити реальну якість моделі, автори зібрали нову вибірку з 3000 облікових записів, яку вручну перевірили. При цьому було два критерія потрапляння до набору: загальна кількість повідомлень не менше 200, кількість повідомлень за останні 3 місяці не менше 90 (в середньому по 1 повідомленню в день). Отримана точність моделі на цьому наборі становила 0.85 AUC, при цьому тренувальним набором даних виступала попередня вибірка. При тренуванні та перевірці тільки з використанням нового набору даних точність становила 0.89 AUC.

Також автори перевірили набори з різними частками прикладів з найпершого тренувального набору та з набору новіших облікових записів. При повному об’єднані наборів та 5-кратній перехресній перевірці точність становила 0.94 AUC. Змінюючи частки прикладів з двох наборів точність варіювалася від 0.90 до 0.94 AUC.

Для того, щоб перевірити значущість різних характеристик облікового запису, автори протестували свій класифікатор ботів окремо на кожній з шести груп

метрик. Отримані результати показали, що найбільш значущими є метрики, що відносяться до користувача та до контенту: точність ідентифікації з використанням метрик цих класів становила більше 0.90 AUC. Інші класи дали точність більше 0.80 AUC.

Помилки першого та другого роду на розширеному наборі даних становили відповідно 15% та 11%.

1.4 Бібліотека TweetBotOrNot

Дана модель ідентифікації ботів у Twitter представлена доцентом Міссурійського Університету Майклом Кірні. Класична реалізація поставляється у вигляді бібліотеки для мови програмування R, але нещодавно автор представив також онлайн сервіс.

В якості алгоритму навчання автор використав метод градієнтного підсилення (англ. Gradient Boosting). Точний об’єм навчальної вибірки не повідомляється. В якості характеристик облікового запису було вибрано більше 100 параметрів, серед них [5]:

а) метрики рівня користувача (англ. user-level attributes): імя, наявність посилання в описі облікового запису, описання, геолокація, дата створення, кількість на частота повідомлень, кількість повідомлень, помічених як «улюблені» іншими користувачами (англ. favorited tweets), публічні списки користувача, кількість друзів та послідовників;

б) шаблони повідомлень верхнього рівня (англ. top-level tweeting patterns): частота, пропорція та часові рамки оригінальних повідомлень (написані самим користувачем, а не зроблені шляхом перепоширення), кількість повідомлень-цитат, перепоширених повідомлень, кількість повідомлень, помічених користувачем як «улюблені» (англ. favorites);

в) текстові шаблони повідомлень (англ. text-based pattern in tweets): кількість згадувань, хештегів, посилань в повідомленнях, довжина тексту повідомлень, пунктуація, складність уживаних слів, тощо.

Модель представлена двома варіаціями: стандартна та швидка.

Стандартна модель оперує всіма вищезазначеними метриками. Відповідно, через велику кількість параметрів, що необхідно обрахувати, для великого набору даних така реалізація може бути повільною. Модель використовує API Twitter та має наступні обмеження: впродовж 15 хвилин можна зробити до 180 оцінок облікових записів.

Швидка варіація використовує тільки метрики рівня користувача та може надавати близько 90 тисяч оцінок за 15 хвилин, але при цьому має меншу точність розпізнавання ботів.

Автор надає наступні показники точності обох варіацій моделі [6]:

- стандартна модифікація:
  а) точність при визначенні ботів: 93.53%;
  б) точність при визначення реальних користувачів: 95.32%;
  в) загальна оцінка точності: 93,8%;
- швидка модифікація:
  а) точність при визначенні ботів: 91.78%;
  б) точність при визначення реальних користувачів: 92.61%;
  в) загальна оцінка точності: 91,9%.

Вихідною оцінкою моделі є число з проміжку [0, 1], яке представляє собою ймовірність того, що обліковий запис керується автоматизованим програмним забезпеченням. Обліковий запис помічається як «бот», якщо вихідна ймовірність приймає значення більше 0.5 (рисунок 1.5) [3].

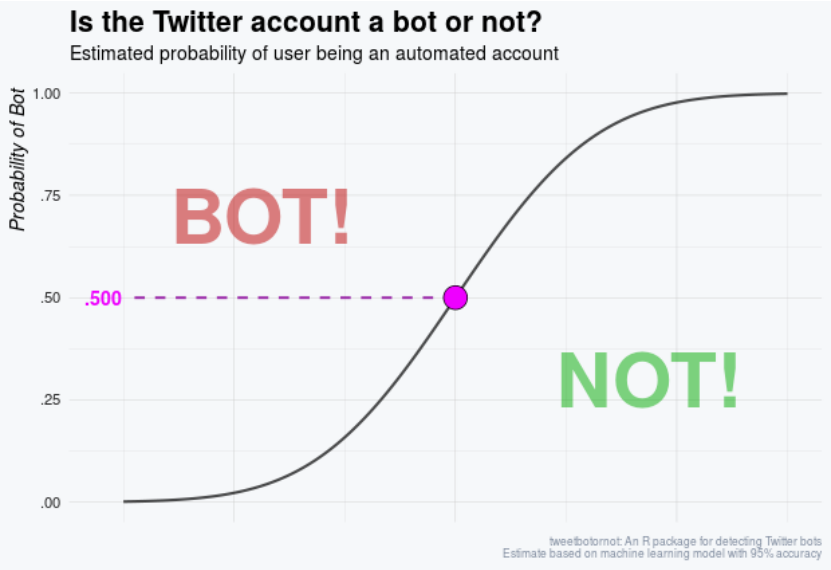

Рисунок 1.5 – Вихідна оцінка облікового запису сервісу TweetBotOrNot.

Висновки до розділу 1

Аналіз поставленої в роботі проблеми показав, що проблема ідентифікації автоматизованих облікових записів у соціальних мережах має великий вплив на інформаційну безпеку в цілому. Протидія маніпуляції свідомості людини та інформаційній пропаганді має вирішальне значення при веденні інформаційних війн. Особливо актуальна дана проблема в умовах розвитку сучасної України як незалежної держави.

В даному розділі було представлено існуючі моделі ідентифікації ботів у соціальній мережі Twitter, наведено їх оцінки точності та значення помилок першого та другого роду.

Дослідження існуючих рішень показало, що точність розпізнавання ботів у Twitter має значні похибки.

2 ПОБУДОВА МОДЕЛІ ІДЕНТИФІКАЦІЇ БОТІВ

В даному розділі детально описано процес створення моделі ідентифікації автоматизованих облікових записів в соціальній мережі Twitter.

Розкрито зміст та сутність навчальної вибірки для моделі. Обґрунтовано вибір класифікуючих параметрів та метри. Наведено теоретичні відомості про використані алгоритми машинного навчання з учителем.

2.1 Постановка задачі

Задача розпізнавання чи є обліковий запис ботом – це задача класифікації машинного навчання.

В задачах такого виду є множина об’єктів, які описуються певним набором параметрів або метрик та розподілені деяким чином по класам (в нашому випадку є два класи: «бот» та «людина»). Задана кінцева множина об’єктів, для яких відомо, до якого класу вони відносяться. Така множина називається вибіркою (або ж навчальний, тренувальний набір). Класова приналежність інших об’єктів невизначена. Задача полягає в побудові алгоритму, який здатен класифікувати будь-який об’єкт з вихідної множини.

Такі алгоритми є відомими та загально уживаними, детально їх розглянуто в розділі 2.4 даної роботи.

Складність задачі класифікації ботів у Twitter полягає в тому, що набір параметрів, який описує конкретний об’єкт (в даному випадку об’єктом виступає обліковий запис користувача), не є заздалегідь відомим та детермінованим. Тобто необхідно спершу знайти такий набір параметрів облікового запису користувача, який зможе з високою точністю встановити приналежність об’єкта до певного класу за конкретно вибраним алгоритмом навчання. При цьому, такий набір метрик

має бути легко відтворюваним, тобто зібрати параметри можна скільки завгодно разів та при будь-яких умовах для будь-якого об’єкта. Вибір параметрів також не повинен залежати від обраного алгоритму навчання. Ця вимога забезпечує можливість використання широкого спектру алгоритмів машинного навчання.

2.2 Формування тренувальної вибірки

Першим етапом при вирішенні поставленої задачі є вибір навчального набору даних. Від якості та повноти навчальної вибірки залежить вибір класифікуючих параметрів, а отже і загальна точність розроблювальної моделі.

На даний момент, у відкритому доступі можна знайти цілий ряд тестових наборів, що складаються з вручну перевірених облікових записів реальних людей та ботів. Але головною проблемою є різнорідність цих наборів.

2.2.1 Аналіз існуючих наборів даних

На першому етапі дослідження було зібрано цілий ряд даних для навчання. Нижче наведено перелік основних джерел тестових вибірок та їх опис:

1) Навчальна вибірка зі змагання по використанню засобів машинного навчання в задачі класифікації ботів Twitter – «NYU Tandon Spring 2017 Machine Learning Competition: Twitter Bot classification» [7]. Складається з 2797 записів та описується наступною множиною стовбців [8]:

- id – числовий тип, унікальній ідентифікатор користувача;
- id_str – строковий тип, строкове представлення поля id;
- screen_name – строковий тип, відображуване імя користувача;

- location – строковий тип, описання геолокації облікового запису в довільній формі;
- description – строковий тип, опис профіля користувача в довільній формі;
- url – строковий тип, посилання на сторонній ресурс в профілі облікового запису;
- followers_count – числовий тип, кількість послідовників;
- friends_count – числовий тип, кількість друзів;
- listed_count – числовий тип, кількість публічних списків, в яких перебуває користувач;
- created_at – строковий тип, дата створення облікового запису;
- favourites_count – числовий тип, кількість повідомлень, помічених як «улюблені»;
- verified – булевий тип, ознака чи є обліковий запис верифікованим;
- statuses_count – числовий тип, кількість повідомлень;
- lang – строковий тип, двозначний код мови користувача;
- status – строковий тип, останнє повідомлення користувача;
- default_profile – булевий тип, якщо «істина» - користувач не змінював стандартну тему фонового малюнка;
- default_profile_image – булевий тип, якщо «істина» - користувач не завантажував власне фото для зображення профілю;
- has_extended_profile – булевий тип, індикатор, що профіль користувача є розширеним;
- name – строковий тип, ім'я користувача;
- bot – числовий тип, число з множини {0, 1}, яке відображає чи є обліковий запис ботом.

2) Вибірка представлена вченими Інституту інформатики і телематики Італійської Національної дослідницької ради в роботі [9]. Загальну вибірку облікових записів розділено на декілька наборів:

- TFP (the fake project) – 100% облікових записів реальних осіб;
- E13 (elections 2013) – 100% облікових записів реальних осіб;
- INT (intertwitter) – 100% ботів;
- FSF (fastfollowerz) – 100% ботів;
- TWT (twittertechnology) – 100% ботів.

Кожен з наборів містить по 4 файла: users, tweets, friends, followers. Нижче наведено відповідні множини стовбців, що описують кожен з файлів (якщо описання не надано, то поле має такий же зміст, як і в попередньому наборі даних):

a) users:

- id – числовий тип;
- name – строковий тип;
- screen_name – строковий тип;
- statuses_count – числовий тип;
- followers_count – числовий тип;
- friends_count – числовий тип;
- favourites_count – числовий тип;
- listed_count – числовий тип;
- created_at – строковий тип;
- url – строковий тип;
- lang – строковий тип;
- time_zone – строковий тип, часовий пояс користувача у вигляді назви міста;
- location – строковий тип;
- default_profile – булевий тип;

- default_profile_image – булевий тип;
- geo_enabled – булевий тип, індикатор чи включено визначення геолокації користувача;
- profile_image_url – строковий тип, посилання на фотографію користувача;
- profile_banner_url – строковий тип, посилання на баннер користувача;
- profile_use_background_image – булевий тип, індикатор чи використовує користувач фонове зображення облікового запису;
- profile_background_image_url_https – строковий тип;
- profile_text_color – строковий тип;
- profile_image_url_https – строковий тип;
- profile_sidebar_border_color – строковий тип;
- profile_background_tile – строковий тип;
- profile_sidebar_fill_color – строковий тип;
- profile_background_image_url – строковий тип;
- profile_background_color- строковий тип;
- profile_link_color – строковий тип;
- utc_offset – числовий тип;
- protected – булевий тип, індикатор чи є обліковий запис закритим для перегляду;
- verified – булевий тип;
- description – строковий тип;
- updated – дата та час, коли було зібрано інформацію про обліковий запис;
- dataset – строковий тип, набір, до якого відноситься обліковий запис.

б) friends:

- source_id – числовий тип, унікальний ідентифікатор користувача, в якого в друзях знаходиться користувач з ідентифікатором target_id;
- target_id – числовий тип, унікальний ідентифікатор користувача, який є другом source_id.

в) followers:

- source_id – числовий тип, ідентифікатор користувача, якого наслідують;
- target_id – числовий тип, ідентифікатор користувача-послідовника користувача з ідентифікатором source_id.

г) tweets:

- created_at – строковий тип, дата створення повідомлення;
- id – числовий тип, унікальний ідентифікатор повідомлення;
- text – строковий тип, текст повідомлення;
- source – строковий тип, посилання на повідомлення;
- user_id – числовий тип, унікальний ідентифікатор користувача-автора повідомлення;
- truncated – булевий тип, індикатор чи було текст обрізано (усі повідомлення в Twitter мають обмеження довжини у 280 символів);
- in_reply_to_status_id – числовий тип, якщо повідомлення це відповідь на інше повідомлення, то це поле містить унікальний ідентифікатор оригінального повідомлення;
- in_reply_to_user_id – числовий тип, якщо повідомлення це відповідь на інше повідомлення, то це поле містить унікальний ідентифікатор автора оригінального повідомлення;

- in_reply_to_screen_name – строковий тип, якщо повідомлення це відповідь на інше повідомлення, то це поле містить відображуване імя автора оригінального повідомлення;
- retweeted_status_id – числовий тип, якщо повідомлення це перепоширення іншого повідомлення, то це поле містить унікальний ідентифікатор оригінального повідомлення;
- geo – масив, містить координати локації, з якої зроблено повідомлення (якщо geo_enabled має значення «істина»);
- place – тип Place, представляє собою окремий документ, який відображає місце асоційоване з повідомленням;
- retweet_count – числовий тип, кількість перепоширень повідомлення;
- reply_count – числовий тип, кількість відповідей на повідомлення;
- favorite_count – числовий тип, кількість разів, коли користувачі додали повідомлення до «улюблених»;
- num_hashtags – числовий тип, кількість хештегів в повідомленні;
- num_urls – числовий тип, кількість посилань в повідомленні;
- num_mentions – числовий тип, кількість згадувань в повідомленні;
- timestamp – дата та час, коли було зібрано інформацію про повідомлення.

3) Вибірка представлена в роботі [10]. Містить окремі групи наборів даних, кожен з яких містить файли users та tweets, вміст такий самий, як і в аналогічних файлах попереднього набору. Перелік та опис груп, що представлені в даному наборі даних:

- Справжні облікові записи (англ. genuine accounts), містить верифіковані облікові записи, що керуються реальними особами.
  - 3 474 облікових записи;
  - 8 377 522 повідомлення;
  - 2011 рік.
- Соціальні спам боти вибірка №1 (англ. social spambots) – облікові записи, що перепоширювали повідомлення італійських політичних кандидатів.
  - 991 обліковий запис;
  - 1 610 176 повідомлень;
  - 2012 рік.
- Соціальні спам боти вибірка №2 – спамери платних застосунків для мобільних пристроїв.
  - 3 457 облікових записи;
  - 428 542 повідомлення;
  - 2014 рік.
- Соціальні спам боті вибірка №3 – спамери продуктів, що продаються на платформі Amazon.com.
  - 464 облікових записи;
  - 1 418 626 повідомлень;
  - 2011 рік.
- Традиційні спам боти вибірка №1 (англ. Traditional spambots) – результати досліджень [11].
  - 1 000 облікових записів;
  - 145 094 повідомлення;
  - 2009 рік.
- Традиційні спам боти вибірка №2 – поширювачі зловмисний посилань.

- 100 облікових записи;
- 74 957 повідомленнь;
- 2014 рік.

- Традиційні спам боти вибірка №3 – автоматизовані облікові записи, що поширювали пропозиції щодо роботи.
  - 433 облікових записи;
  - 5 794 931 повідомлення;
  - 2013 рік.
- Традиційні спам боти вибірка №4 – інша група ботів, що поширювали пропозиції щодо роботи.
  - 1 128 облікових записи;
  - 133 311 повідомлення;
  - 2009 рік.
- Фальшиві послідовники (англ. fake followers) – облікові записи, що створенні заради збільшення кількості послідовників іншого облікового запису.
  - 3 351 обліковий запис;
  - 196 027 повідомлень;
  - 2012 рік.

4) Набір характеристик з досліджень [12]. Вибірка описується наступним набором параметрів:

- screen_name – строковий тип;
- user_statuses – числовий тип, загальна кількість повідомлень користувача;
- user_tweets – числовий тип, кількість власних повідомлень;
- user_retweets – числовий тип, кількість перепоширень;
- user_favourites – числовий тип, кількість повідомлень, що додані користувачем до «улюблених»;

- user_replies_and_mentions – числовий тип, кількість відповідей та згадувань іншими користувачами;
- likes_per_tweet – числовий тип, середня кількість уподобань користувачів на повідомленння;
- retweets_per_tweet – числовий тип, середня кількість перепоширень на повідомлення;
- lists_account_age_ratio – числовий тип, відношення кількості публічних списків до віку облікового запису;
- follower_friend_ratio – числовий тип, відношення кількості послідовників до кількості друзів;
- lifetime_statuses_freq – числовий тип, частота повідомлень упродовж всього періоду існування облікового запису;
- favourite_tweet_ratio – числовий тип, відношення кількості разів, що повідомлення було додано до «улюблених» іншими користувачами до кількості повідомлень;
- age_of_account_in_days – числовий тип, вік облікового запису у днях;
- sources_count – числовий тип, кількість різних джерел, звідки були написані повідомлення (наприклад, iPhone, Android, Web, тощо);
- urls_count – числовий тип, кількість посилань в повідомленнях;
- daily_favouriting_frequency – числовий тип, середня кількість додавань до «улюбленого» за день.

5) Набір класифікацій з досліджень [13]. Цей набір описується такими параметрами:

- screen_name – строковий тип;
- user_tweeted – числовий тип, кількість повідомлень;
- user_retweeted – числовий тип, кількість перепоширень;

- user_favourited – числовий тип, кількість повідомлень, помічених як «улюблені»;
- user_replied – числовий тип, кількість повідомлень-відповідей;
- likes_per_tweet – числовий тип, середня кількість уподобань на повідомлення;
- retweets_per_tweet – числовий тип, середня кількість перепоширень на повідомлення;
- lists_per_user – числовий тип, кількість публічних списків користувача;
- follower_friend_ratio – числовий тип, відношення кількості послідовників до кількості друзів;
- tweet_frequency – числовий тип, частота повідомлень;
- favourite_tweet_ratio – числовий тип;
- age_of_account_in_days – числовий тип;
- sources_count – числовий тип;
- urls_count – числовий тип;
- cdn_content_in_kb – числовий тип, обсяг даних у кілобайтах, що був розміщений користувачем;
- source_identity – масив строкових типів, назви джерел повідомлень.

6) Набір верифікованих облікових записів з [14]. Параметри цього набору повністю співпадають з відповіддю API Twitter на запит інформації про користувача. Користувачами з цієї вибірки є, переважно, публічні люди: відомі актори, музиканти, зірки спорту, тощо. Цей набір було розглянуто для врахування крайових випадків, описаних в розділі 1 даної роботи.

Як видно з аналізу існуючих джерел тренувальних наборів для розпізнавання ботів у Twitter, багато даних носять застарілий характер та мають давність більше 5 років, що не може бути використано для побудови сучасної та якісної моделі

ідентифікації. Окрім цього, з опису даних можна побачити, що вони в більшості своїй різнорідні, за винятком наборів, що були отримані одними й тим ж самими авторами.

Беручи до уваги ці обставини, було вирішено зібрати заново інформацію про облікові записи з наявних наборів. Такий підхід дозволить мати найновіші дані про користувачів, але в той же час інформацію про те, чи є обліковий запис автоматизованим, не буде втрачено. Крім цього, ряд наборів стане повнішим з точки зору інформації, що може бути отримана за допомогою API Twitter.

2.2.2 Збір даних соціальної мережі Twitter

Для збору даних про користувача та його повідомлення було використано мову програмування Python та бібліотеку для роботи з API Twitter – tweepy.

Щоб стати споживачем інформації, що віддає API Twitter, потрібно отримати відповідні ключі та маркери доступу, які складаються з двох пар:

- атрибути споживача: consumer_key та consumer_secret;
- атрибути доступу: access_token та access_token_secret**.**

Першим кроком в процесі оновлення тренувальних наборів даних стало написання програмного забезпечення, що виконує збір, обробку та зберігання даних. Створене програмне забезпечення має наступні структурні частини з відповідним описом:

- __init__.py – пустий допоміжний файл для ініціалізації директорії як програмного модуля;
- multiprocess_scraping.py – модуль логіки багатопроцесорного збору інформації з API (рисунок 2.1);

- adapter.py – адаптер бібліотеки tweepy, основною метою якого є інкапсуляція ключів та маркерів доступу, а також первинна обробка результатів запитів до API (рисунок 2.2);
- settings.py – модуль, що містить налаштування програми, зокрема логіку отримання ключів доступу зі змінних оточення (рисунок 2.3).

Слід відмітити, що API Twitter має обмеження кількості запитів: упродовж 15 хвилин можна зробити 900 запитів на отримання інформації про користувача та його повідомлення. Багатопроцесорний збір даних будується на наявності множини ключів доступу певного обсягу (при виконанні даної роботи використовувався набір зі 167 валідних квартетів ключів та маркерів доступу).

```
import sys

from multiprocessing.dummy import Pool as ThreadPool

from funcy import compact, flatten

from utils import chunk_list, get_data
from .adapter import EncodeTwitterError, TwitterAdapter


THREAD_COUNT = 20


def scrape_users(names, credentials):
    twitter = TwitterAdapter(credentials)
    return [twitter.get_user(name) for name in names]


def scrape_tweets(names, credentials):
    twitter = TwitterAdapter(credentials)
    tweets = []
    for name in names:
        try:
            user_tweets = twitter.get_tweets(name)
        except EncodeTwitterError:
            pass
        else:
            tweets.append(user_tweets)
    return tweets


def scrape_twitter(names, credentials_file, mode='scrape_users'):
    credentials = get_data(credentials_file)
    func = getattr(sys.modules[__name__], mode)
    threads_count = min(len(credentials), len(names), THREAD_COUNT)
    names_chunks = list(chunk_list(names, threads_count))
    pool = ThreadPool(threads_count)
    results = pool.starmap(func, zip(names_chunks, credentials))
    return compact(list(flatten(results)))
```

Рисунок 2.1 – Програмний код модуля багатопроцесорного збору даних

```python
import time

import tweepy

from funcy import omit, retry

from utils import to_ts
from .settings import TWITTER_CREDENTIALS


class EncodeTwitterError(Exception):
    """Exception for json Failed to parse JSON payload error."""


class TwitterAdapter:
    def __init__(self, credentials=TWITTER_CREDENTIALS):
        auth = tweepy.OAuthHandler(
            consumer_key=credentials['consumer_key'],
            consumer_secret=credentials['consumer_secret']
        )
        auth.set_access_token(
            key=credentials['access_token'],
            secret=credentials['access_token_secret']
        )
        self.api = tweepy.API(
            auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True
        )

    def get_user(self, user_name, omit_fields=None):
        try:
            user = self.api.get_user(user_name)._json
        except tweepy.TweepError:
            pass
        else:
            user['scrape_date'] = int(time.time())
            user['created_at'] = to_ts(
                user['created_at'], date_format='%a %b %d %H:%M:%S +0000 %Y'
            )
            if omit_fields:
                user = omit(user, omit_fields)
            return user

    @retry(5, EncodeTwitterError)
    def get_tweets(self, user_name, count=200):
        try:
            tweets = self.api.user_timeline(screen_name=user_name, count=count)
        except tweepy.TweepError as error:
            if 'Not authorized' not in error.reason:
                raise EncodeTwitterError
        else:
            return [tweet._json for tweet in tweets]
```

Рисунок 2.2 – Програмний код адаптера бібліотеки tweepy

```
import os


ENV = os.environ.get

TWITTER_CREDENTIALS = {
    'consumer_key': ENV('TW_CONSUMER_KEY'),
    'consumer_secret': ENV('TW_CONSUMER_SECRET'),
    'access_token': ENV('TW_ACCESS_TOKEN'),
    'access_token_secret': ENV('TW_ACCESS_TOKEN_SECRET'),
}
```

Рисунок 2.3 – Програмний код модуля налаштувань

Для того, щоб зібрати нові дані про облікові записи в знайдених тренувальних наборах, всі розмічені вибірки було злито в одну. З отриманого єдиного розширеного набору було взято лише один параметр облікового запису – screen_name, якого виявилося достатньо для запиту до API Twitter, щоб отримати повну інформацію про користувача. З вихідних наборів також був збережений індикатор, чи є обліковий запис ботом.

Об’єм розширеної вибірки становив більше 50 000 облікових записів (при цьому з множини верифікованих користувачів було взято 5 000 облікових записів). Процес збирання інформації про цих користувачів було організовано у вигляді часток по 5 000 запитів. В результаті збору вдалося отримати вибірку з 21 275 облікових записів, серед яких 5 728 ботів та 15 547 користувачів (серед них 4 397 верифікованих). Слід зазначити, що частка ботів та реальних людей у початковому об’єднаному наборі була майже однаковою, але в процесі збору даних виявилося, що більшість облікових записів ботів було заблоковано адміністрацією Twitter.

Отриманий тренувальний набір складався з наступних параметрів (значення більшості з них було розкрито в попередньому розділі) (рисунок 2.4):

- bot – булеве значення, індикатор того, що обліковий запис помічено як бота;
- contributors_enabled – застарілий атрибут, завжди null;
- created_at – строковий тип;
- default_profile – булевий тип;

```
{
  "bot": 0,
  "contributors_enabled": false,
  "created_at": 1387802401,
  "default_profile": false,
  "default_profile_image": false,
  "description": "jers//glasgow",
  "entities": {
    "description": {
      "urls": []
    }
  },
  "favourites_count": 11364,
  "follow_request_sent": false,
  "followers_count": 310,
  "following": false,
  "friends_count": 191,
  "geo_enabled": true,
  "has_extended_profile": true,
  "id": 2259016411,
  "id_str": "2259016411",
  "is_translation_enabled": false,
  "is_translator": false,
  "lang": "en",
  "listed_count": 1,
  "location": "Channel Islands",
  "name": "chloë",
  "notifications": false,
  "profile_background_color": "C0DEED",
  "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
  "profile_background_image_url_https":
"https://abs.twimg.com/images/themes/theme1/bg.png",
  "profile_background_tile": true,
  "profile_banner_url": "https://pbs.twimg.com/profile_banners/2259016411/1530128948",
  "profile_image_url":
"http://pbs.twimg.com/profile_images/1012059461282918404/tPa5LesZ_normal.jpg",
  "profile_image_url_https":
"https://pbs.twimg.com/profile_images/1012059461282918404/tPa5LesZ_normal.jpg",
  "profile_link_color": "422DCC",
  "profile_location": null,
  "profile_sidebar_border_color": "FFFFFF",
  "profile_sidebar_fill_color": "DDEEF6",
  "profile_text_color": "333333",
  "profile_use_background_image": true,
  "protected": false,
  "scrape_date": 1555958284,
  "screen_name": "ChloeWalker98",
  "status": null,
  "statuses_count": 4892,
  "time_zone": null,
  "translator_type": "none",
  "url": null,
  "utc_offset": null,
  "verified": false,
  "withheld_in_countries": null
}
```


Рисунок 2.4 – Приклад відповіді API Twitter на запит інформації про користувача

- default_profile_image – булевий тип;
- description – строковий тип;
- entities – об’єкт типу Entities;
- favourites_count – числовий тип;
- follow_request_sent – застарілий атрибут, завжди null;
- followers_count – числовий тип;
- following – застарілий атрибут, завжди null;
- friends_count – числовий тип;
- geo_enabled – застарілий атрибут, завжди null;
- has_extended_profile – застарілий атрибут, завжди null;
- id – числовий тип;
- id_str – строковий тип;
- is_translation_enabled – застарілий атрибут, завжди null;
- is_translator – застарілий атрибут, завжди null;
- lang – застарілий атрибут, завжди null;
- listed_count – числовий тип;
- location – строковий тип;
- name – строковий тип;
- notifications – застарілий атрибут, завжди null;
- profile_background_color – застарілий атрибут, завжди null;
- profile_background_image_url – застарілий атрибут, завжди null;
- profile_background_image_url_https – застарілий атрибут, завжди null;
- profile_background_tile – застарілий атрибут, завжди null;
- profile_banner_url – строковий тип;
- profile_image_url – застарілий атрибут, завжди null;
- profile_image_url_https – строковий тип;
- profile_link_color – застарілий атрибут, завжди null;

- profile_location – застарілий атрибут, завжди null;
- profile_sidebar_border_color – застарілий атрибут, завжди null;
- profile_sidebar_fill_color – застарілий атрибут, завжди null;
- profile_text_color - – застарілий атрибут, завжди null;
- profile_use_background_image – застарілий атрибут, завжди null;
- protected – булевий тип;
- scrape_date – дата та час збору інформації;
- screen_name – строковий тип;
- status – об’єкт типу Tweet;
- statuses_count – числовий тип;
- time_zone – застарілий атрибут, завжди null;
- translator_type – застарілий атрибут, завжди null;
- url – строковий тип;
- utc_offset – застарілий атрибут, завжди null;
- verified – булевий тип;
- withheld_in_countries – масив строкових типів, список країн, контент з яких вилучено з новинної стрічки користувача.

Отримана інформація зберігалася у форматі JSON.

Після отримання інформації щодо користувачів, було виконано пошук їх повідомлень. API Twitter дозволяє отримати за один запит до 200 найновіших повідомлень користувачів. Всього було зібрано 4 371 303 повідомлень для відповідного набору користувачів. Повідомлення збиралися також по частках: всього було зібрано 7 часток загальним об’ємом 11,3 гігабайта.

Отриманий набір повідомлень користувачів складається з наступних параметрів (значення деяких параметрів було розкрито в попередньому розділі) (рисунок 2.5):

```
{"created_at": "Tue Nov 25 16:23:26 +0000 2014",
 "id": 537280409823096832,
 "id_str": "537280409823096832",
 "text": "Just started my INSANITY: 60-Day Total Body Conditioning Workout DVD Program
- wish me luck!",
 "truncated": false,
 "entities": {"hashtags": [],
  "symbols": [],
  "user_mentions": [],
  "urls": [{"url": "http://t.co/5wNQF8F4kW",
    "expanded_url":
"http://www.amazon.com/gp/product/B002QZ1RS6/ref=as_li_tf_tl?ie=UTF8&camp=1789&creativ
e=9325&creativeASIN=B002QZ1RS6&linkCode=as2&tag=activepubs",
    "display_url": "amazon.com/gp/product/B00…",
    "indices": [94, 116]}]},
 "source": "<a href='tweetadder.com' rel='nofollow'>TweetAdder v4</a>",
 "in_reply_to_status_id": null,
 "in_reply_to_status_id_str": null,
 "in_reply_to_user_id": null,
 "in_reply_to_user_id_str": null,
 "in_reply_to_screen_name": null,
 "user": "HORSESVIBE",
 "geo": null,
 "coordinates": null,
 "place": null,
 "contributors": null,
 "is_quote_status": false,
 "retweet_count": 0,
 "favorite_count": 0,
 "favorited": false,
 "retweeted": false,
 "possibly_sensitive": false,
 "lang": "en"}
```


Рисунок 2.5 – Приклад відповіді API Twitter на запит повідомлень користувача

- created_at – строковий тип;
- id – числовий тип;
- id_str – строковий тип;
- text – строковий тип;
- truncated – булевий тип;
- entities – об’єкт класу Entities;
- source – строковий тип;
- in_reply_to_status_id – числовий тип;
- in_reply_to_status_id_str – строковий тип;
- in_reply_to_user_id – числовий тип;
- in_reply_to_user_id_str – строковий тип;

- in_reply_to_screen_name – строковий тип;
- user – об’єкт типу User;
- geo – застарілий атрибут, завжди null;
- coordinates – об’єкт типу Coordinates;
- place – об’єкт типу Places;
- contributors – застарілий атрибут, завжди null;
- is_quote_status – булевий тип;
- retweet_count – числовий тип;
- favorite_count – числовий тип;
- favorited – булевий тип, індикатор чи було повідомлення додано до «улюбленого»;
- retweeted – булевий тип, індикатор чи було повідомлення перепоширено;
- possibly_sensitive – булевий тип, показник того, що посилання в повідомленні може містити делікатну інформацію;
- lang – строковий тип.

Формат зберігання повідомлення – JSON.

Після того, як були зібрані повні дані про користувачів та їх повідомлення, постала задача відбору параметрів, які будуть відображати об’єкт облікового запису при машинному навчанні.

2.3 Вибір класифікуючих метрик

Відбір параметрів, що будуть представляти обліковий запис користувача було почато з находження закономірностей між отриманими параметрами та класом об’єкта. Всі параметри було інтерпретовано як числа. Наприклад,

параметри булевих типів були представлені як 1 або 0. Строкові типи даних були перетворені на значення довжин.

Весь тренувальний набір було розбито на 3 групи: боти, реальні користувачі, верифіковані користувачі. Такий розподіл класу не автоматизованих облікових записів було зроблено через відмінності між способом ведення облікового запису звичайними людьми та публічними.

Було проаналізовано залежність класу облікового запису від числових параметрів. Нижче наведено декілька прикладів результатів аналізу:

- кількість друзів (рисунки 2.6, 2.7 та 2.8);
- кількість послідовників (рисунки 2.9 та 2.10);
- кількість публічних списків користувача (рисунки 2.11 та 2.12);
- кількість повідомлень (рисунки 2.13, 2.14 та 2.15);
- кількість повідомлень, що відмічені користувачем як «улюблені» (рисунки 2.16, 2.17 та 2.18);
- довжина відображуваного імені (рисунок 2.19);

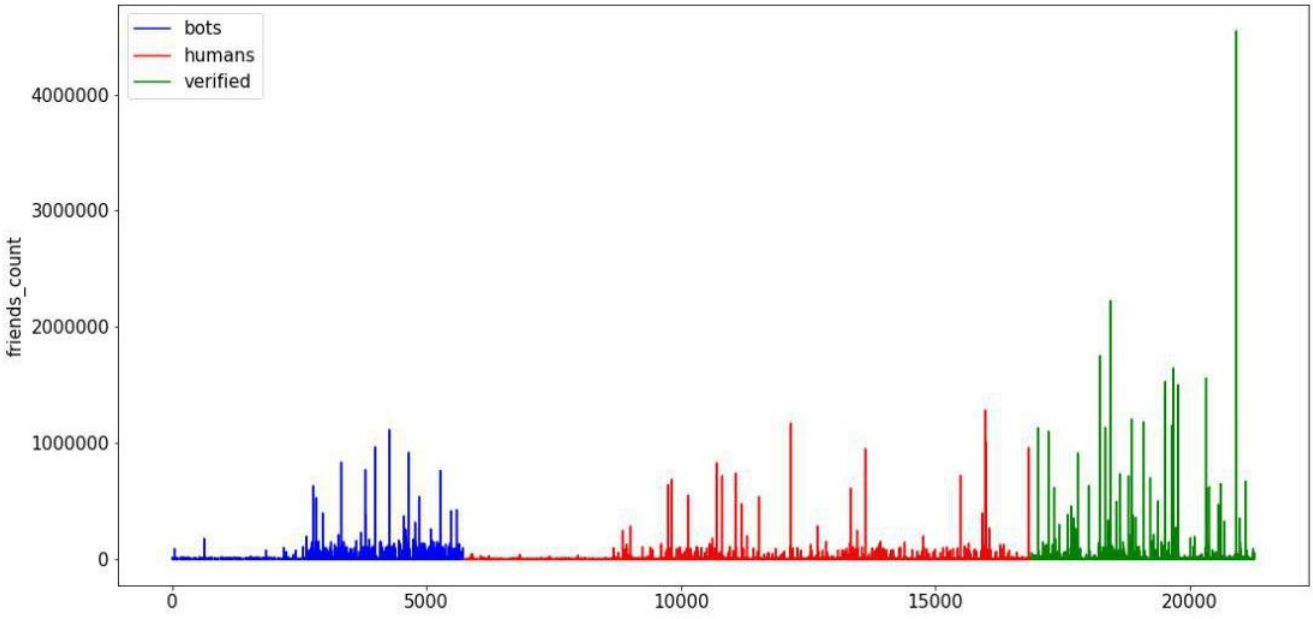

Рисунок 2.6 – Кількість друзів ботів, реальних користувачів та верифікованих користувачів

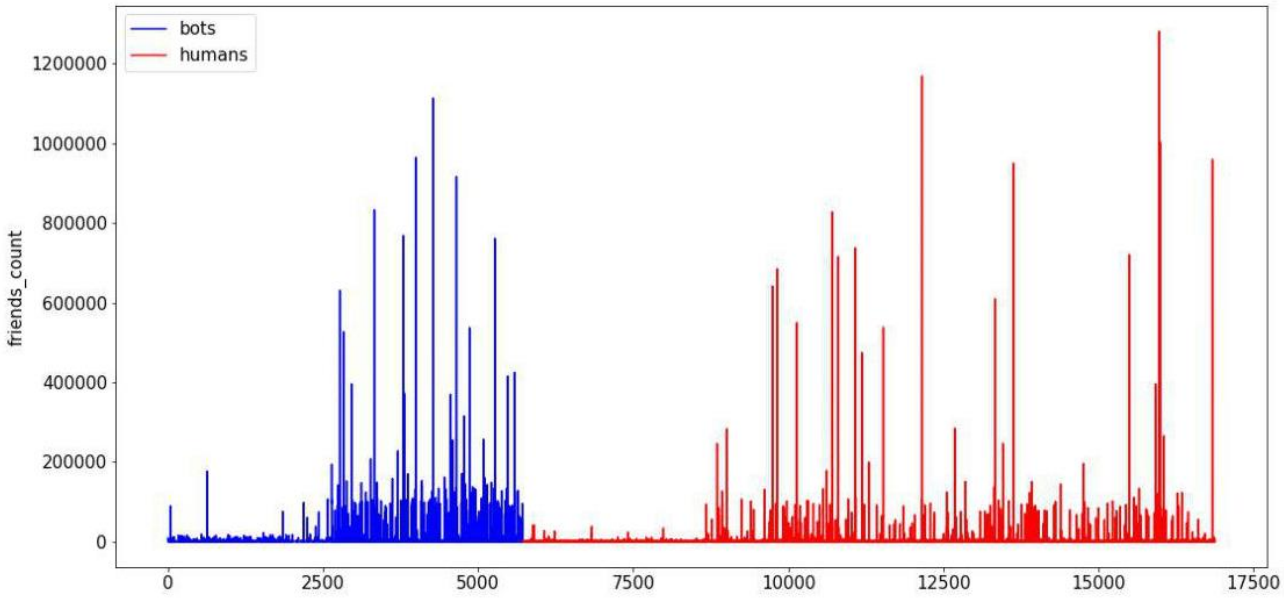

Рисунок 2.7 – Кількість друзів ботів та реальних користувачів

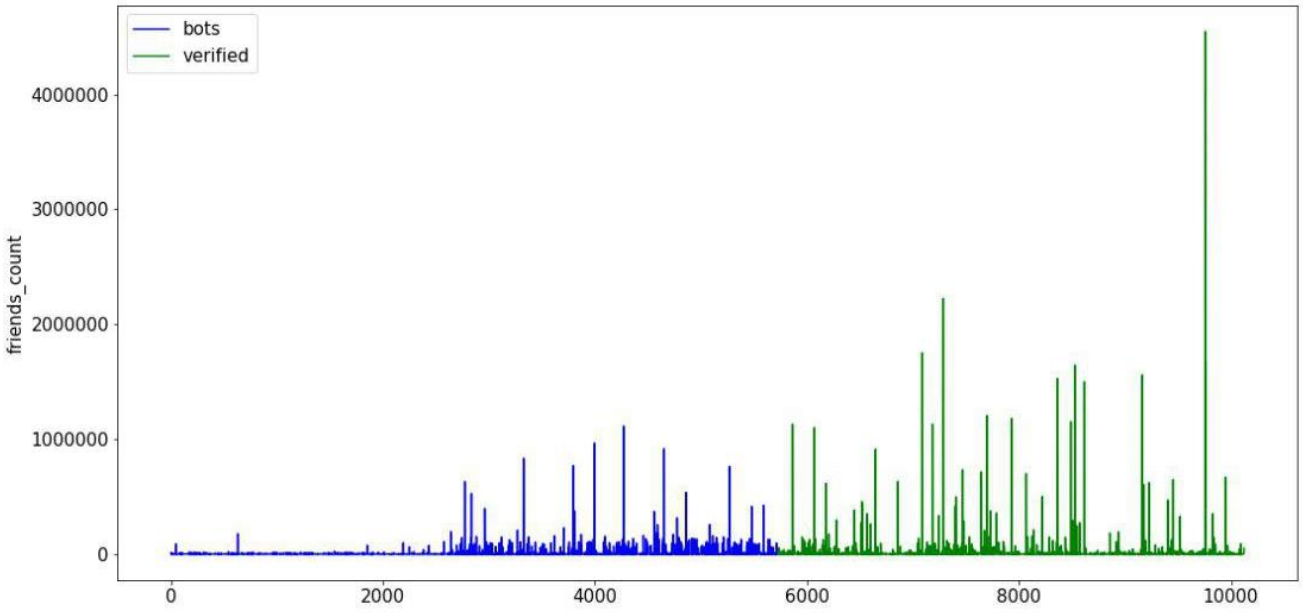

Рисунок 2.8 – Кількість друзів ботів та верифікованих користувачів

Як видно з наведених графіків, спостерігаються наступні закономірності: максимальні значення кількості друзів мають реальні користувачі, при цьому, піки значень для верифікованих облікових записів значно перевищують максимальні значення для звичайних користувачів. Проте, коливання кількості друзів у ботів є меншими (тобто, діапазон значень щільніший).

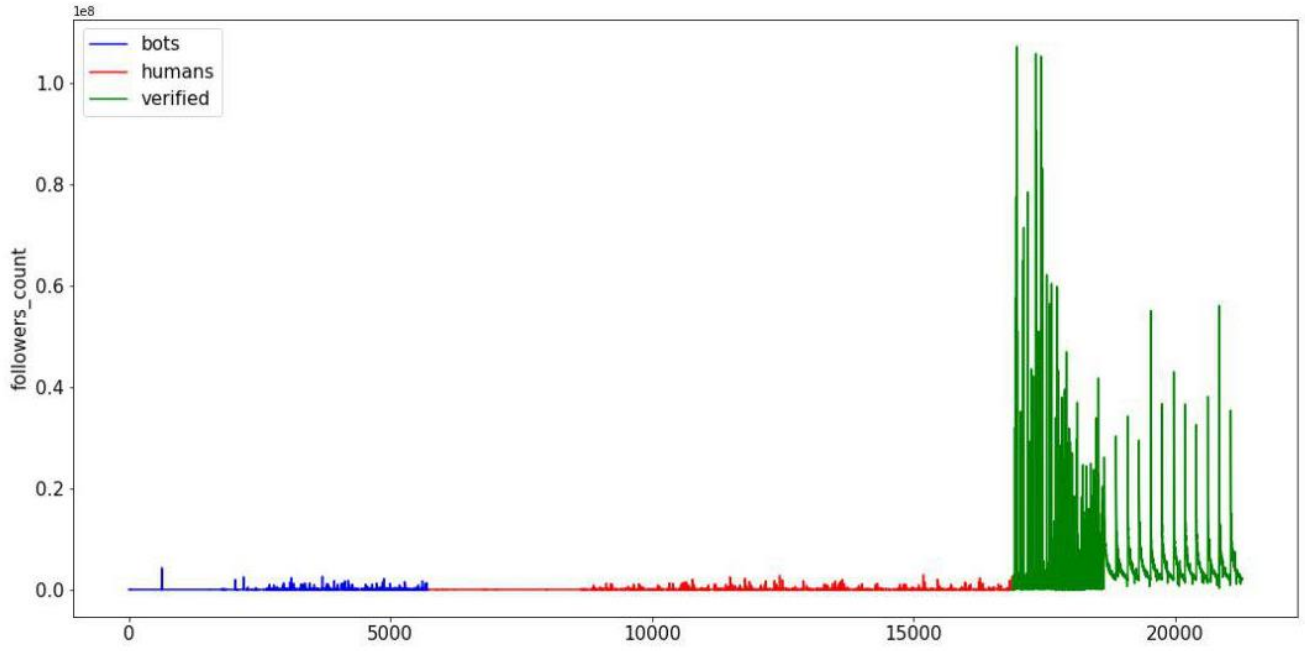

Рисунок 2.9 – Кількість послідовників ботів, реальних користувачів та верифікованих користувачів

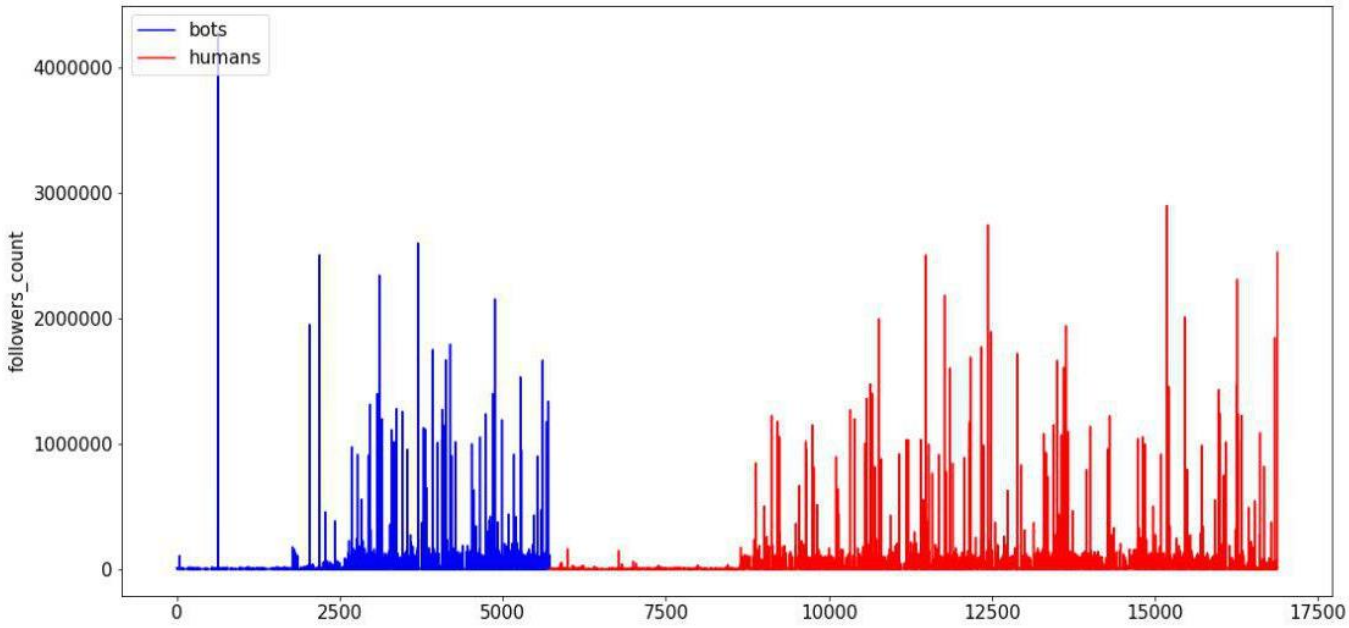

Рисунок 2.10 – Кількість послідовників ботів та реальних користувачів

Очевидною тенденцією є великі показники для облікових записів відомих людей. Якщо порівнювати звичайних користувачів та ботів, то знову спостерігається менший розкид значень для облікових записів ботів.

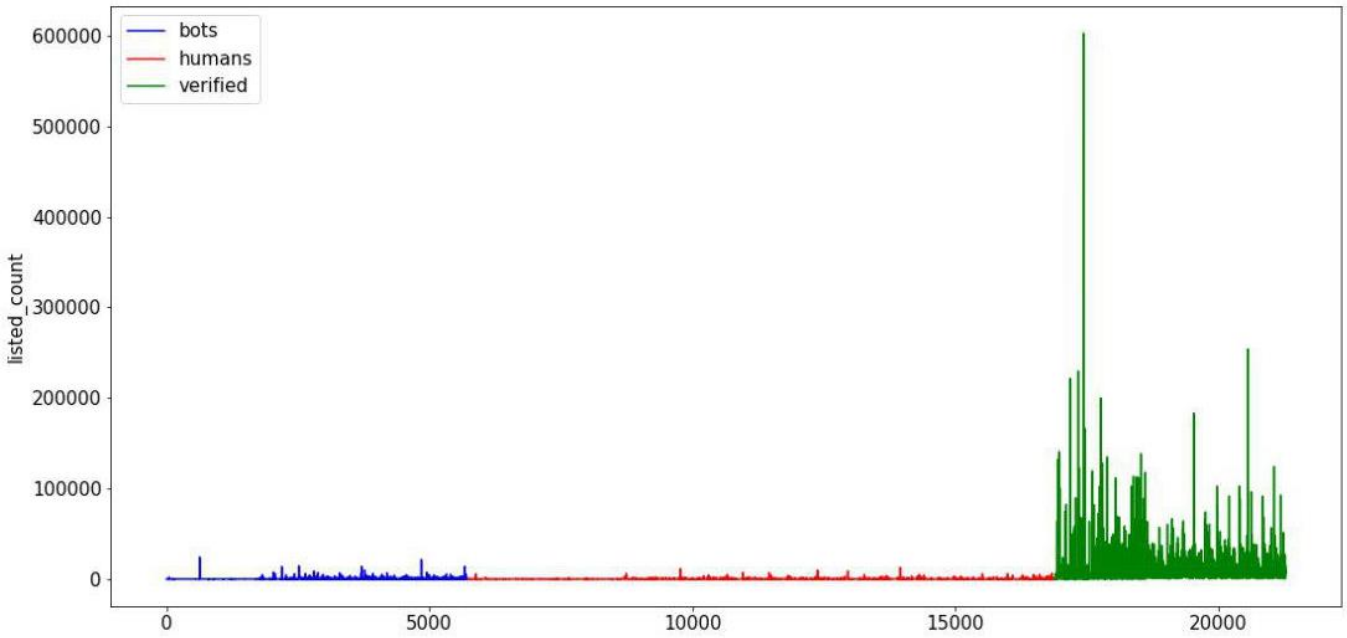

Рисунок 2.11 – Кількість публічних списків ботів, реальних користувачів та верифікованих користувачів

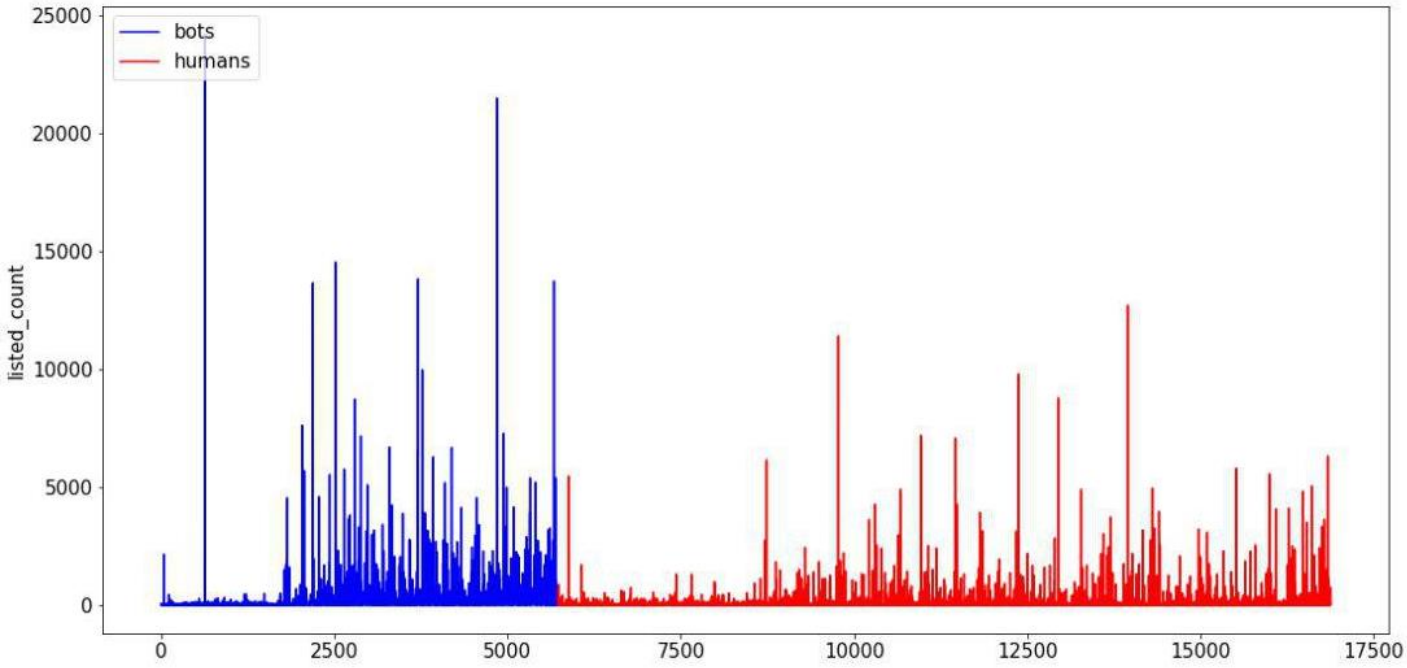

Рисунок 2.12 – Кількість публічних списків ботів та реальних користувачів

Найбільші показники належать групі верифікованих облікових записів. Можна зробити висновок, що якщо відкинути пікові значення, то більша кількість публічних списків може бути ознакою того, що обліковий запис є ботом.

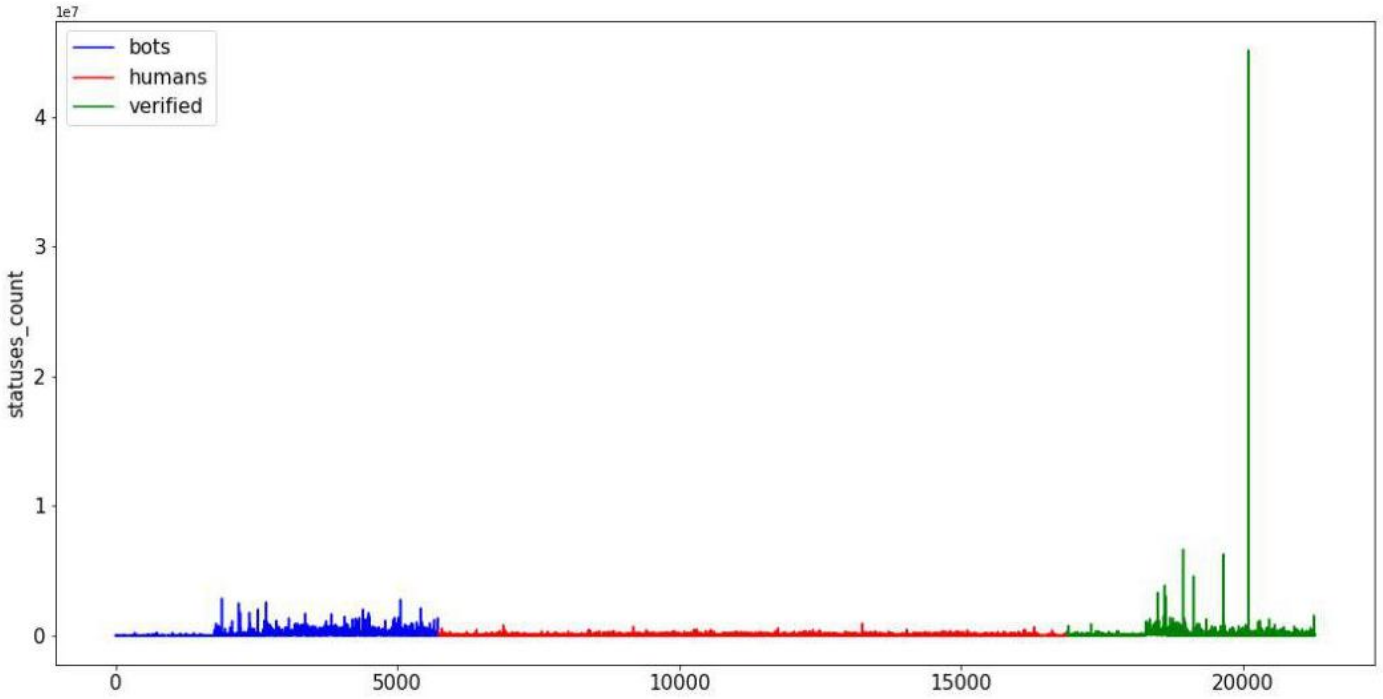

Рисунок 2.13 – Кількість повідомлень ботів, реальних користувачів та верифікованих користувачів

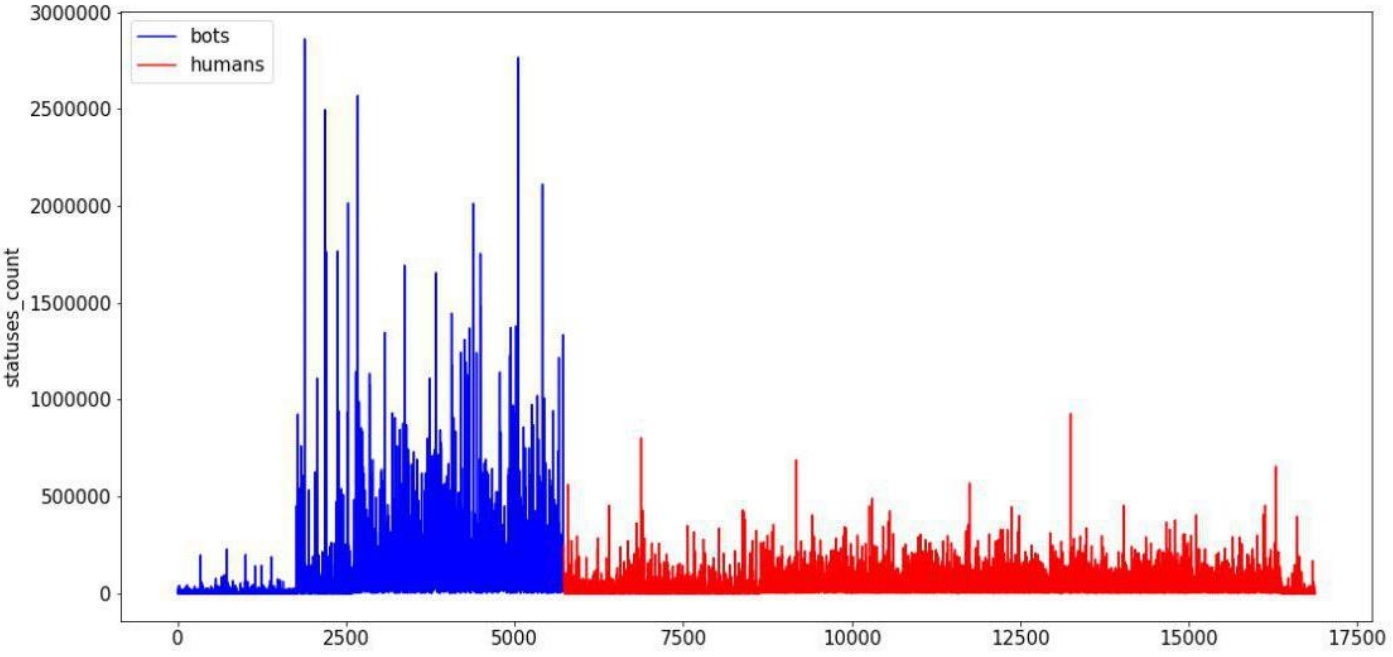

Рисунок 2.14 – Кількість повідомлень ботів та реальних користувачів

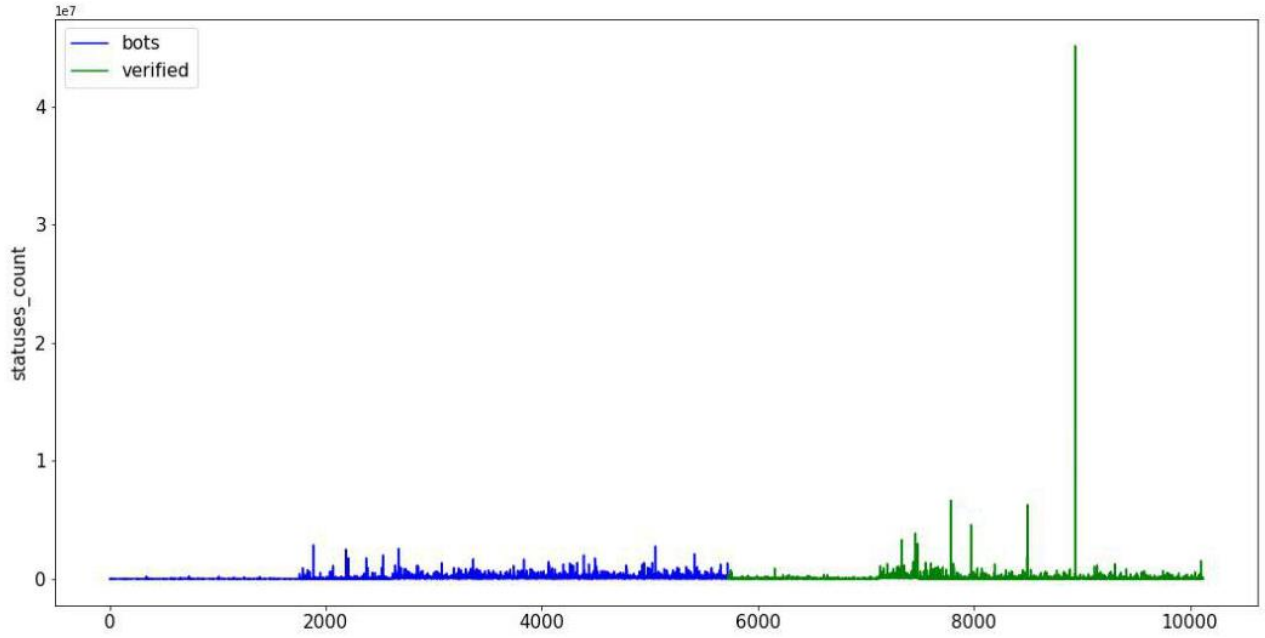

Рисунок 2.15 – Кількість повідомлень ботів та верифікованих користувачів

Прослідковується явна залежність кількості повідомлень від класу облікового запису: боти демонструють поведінку забруднювачів інформаційного середовища, показники кількості їх повідомлень перевищують навіть облікові записи публічних людей з мільйонною аудиторією. Можна вважати дану метрику, як одну з вирішальних при класифікації облікового запису.

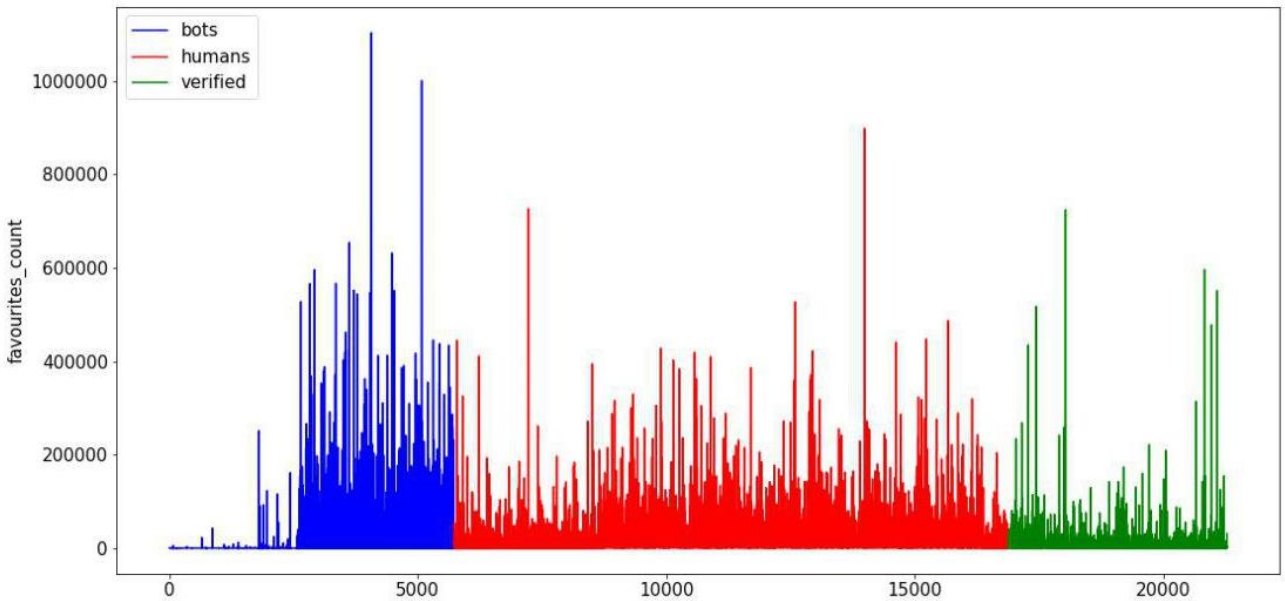

Рисунок 2.16 – Кількість вподобань ботів, реальних та верифікованих користувачів

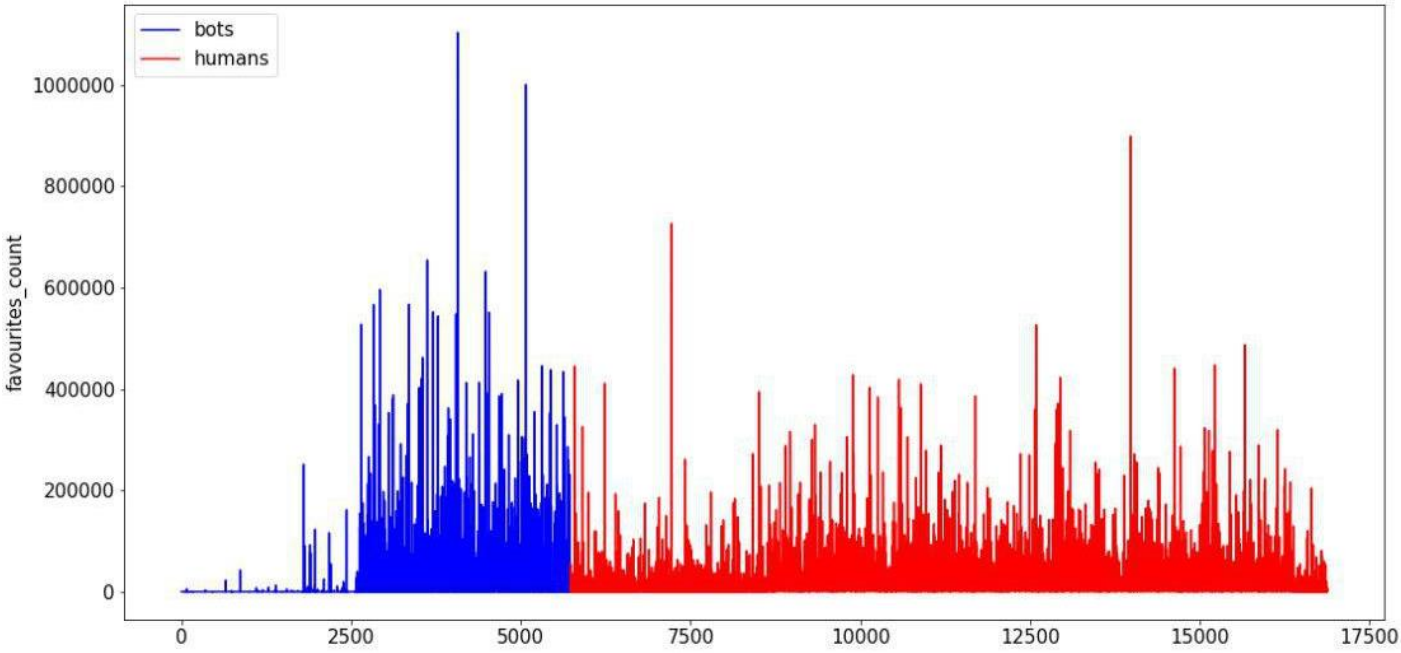

Рисунок 2.17 – Кількість вподобань ботів та реальних користувачів

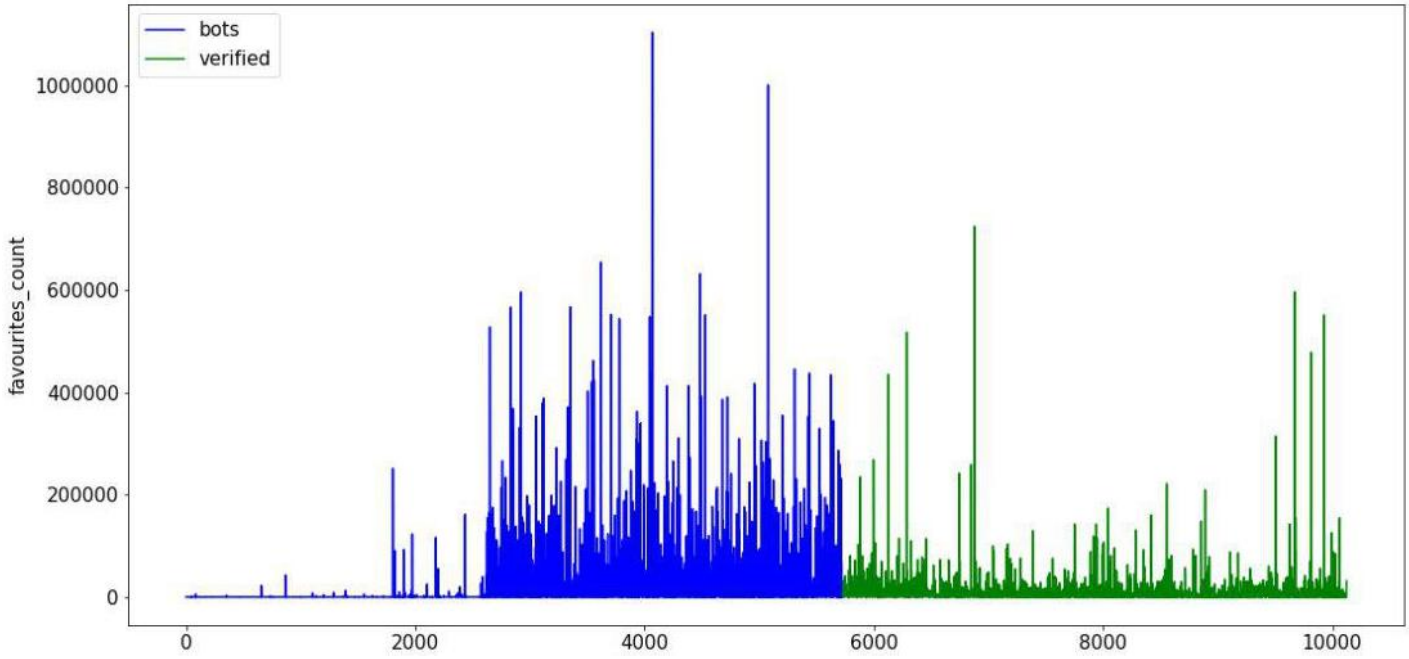

Рисунок 2.18 – Кількість вподобань ботів та верифікованих користувачів

Спостерігаються тенденції характерні для кількості повідомлень – кількість вподобань ботів суттєво вища за аналогічні показники реальних користувачів. Найнижчі показники притаманні верифікованим користувачам. Дану метрику також можна вважати як ту, що має значимість при класифікації облікових записів.

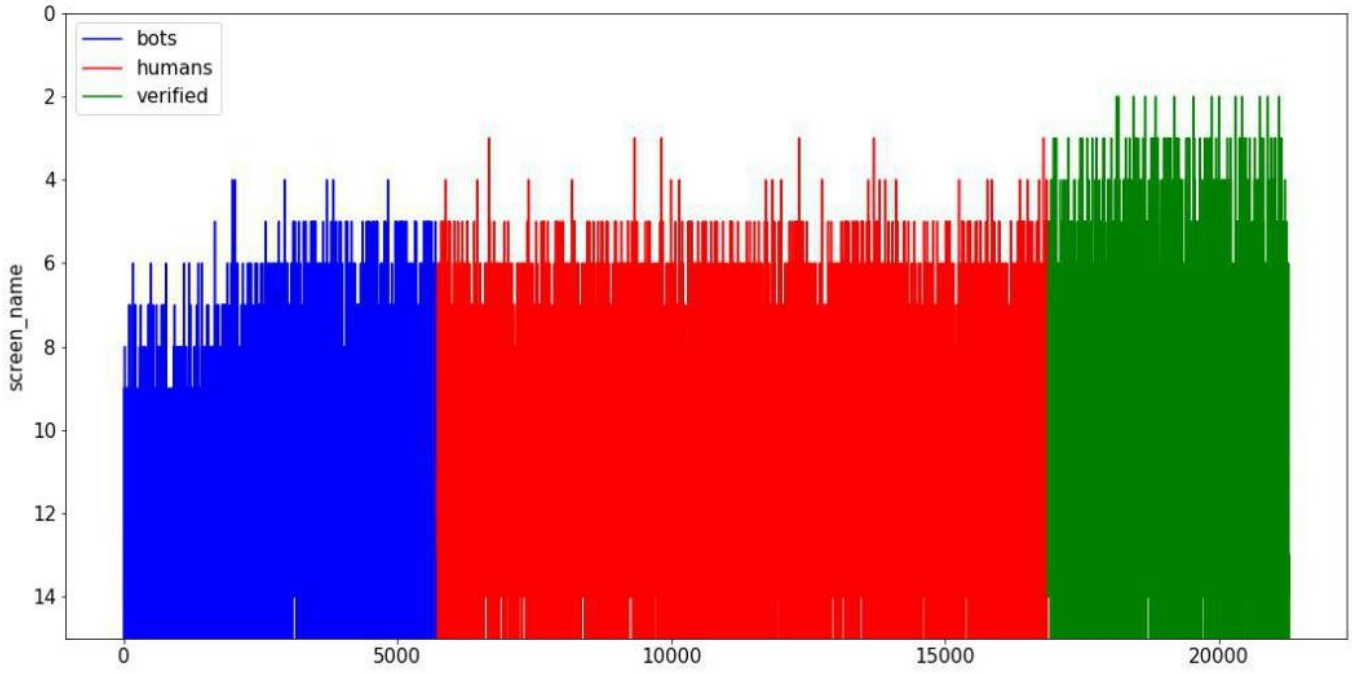

Рисунок 2.19 – Довжина відображуваного імені ботів, реальних користувачів та верифікованих користувачів

Помітна наступна закономірність: мінімальні показники довжини відображуваного імені мають облікові записи ботів, максимальні – верифікованих користувачів.

Після побудови графіків залежності різних параметрів облікових записів від класу, було виділено показники, що виявили найпомітніші закономірності. Результати дослідження підтвердили гіпотезу авторів моделі Botometer – найбільш значущими при класифікації облікового запису є метрики пов’язані з користувачем та контентом, що генерує користувач [1]. В подальших етапах роботи перевірялися тільки метрики такого виду.

Після вибору остаточних параметрів навчальної вибірки моделі всі зібрані облікові записи та їх повідомлення було трансформовано у числові вектори з координатами, що представляють значення обраних метрик. Програмний код трансформацій наведено на рисунках 2.20 та 2.21.

```
def transform_user(user):
    account_age_days = (
        (user['scrape_date'] - user['created_at']) // SECONDS_IN_DAY
    )
    return {
        'account_age_days': account_age_days,
        'bot': user['bot'],
        'default_profile': user['default_profile'],
        'default_profile_image': user['default_profile_image'],
        'description_length': len(user.get('description', '')),
        'description_url_present': (
            bool(get_in(user['entities'], ['description', 'urls']))
        ),
        'favourites_count': user['favourites_count'],
        'followers_count': user['followers_count'],
        'friends_count': user['friends_count'],
        'listed_count': user['listed_count'],
        'location_present': bool(user.get('location')),
        'name_length': len(user.get('name', '')),
        'protected': user['protected'],
        'screen_name': user['screen_name'],
        'screen_name_length': len(user['screen_name']),
        'statuses_count': user['statuses_count'],
        'tweets_per_day': user['statuses_count'] / account_age_days,
        'url_present': bool(user.get('url')),
        'verified': user['verified'],
    }
```

Рисунок 2.20 – Програмний код трансформації параметрів облікового запису в метрики навчальної вибірки

```
def transform_tweet(tweet):
    return {
        'hashtags': len(get_in(tweet, ['entities', 'hashtags'], default=[])),
        'is_retweet': bool(tweet.get('retweeted_status')),
        'media': len(get_in(tweet, ['entities', 'media'], default=[])),
        'polls': len(get_in(tweet, ['entities', 'polls'], default=[])),
        'symbols': len(get_in(tweet, ['entities', 'symbols'], default=[])),
        'text_length': len(tweet['text']),
        'screen_name': tweet['user'],
        'user_mentions': len(
            get_in(tweet, ['entities', 'user_mentions'], default=[])
        ),
        'urls': len(get_in(tweet, ['entities', 'urls'], default=[])),
    }
```

Рисунок 2.21 – Програмний код трансформації параметрів повідомлення в метрики навчальної вибірки

Крім трансформації, параметри повідомлень були агреговані у середні та сумарні значення (рисунок 2.22) для отримання повного розуміння тенденцій показників при загальній картині, а не для окремого повідомлення. Завершальним

етапом підготовки тренувального набору даних стало злиття метрик облікового запису та метрик відповідних повідомлень (рисунок 2.23).

```
def aggregate_tweets(tweets):
    tweets_groups = []
    tweets = sorted(tweets, key=lambda x: x['screen_name'])
    for _, group in groupby(tweets, lambda x: x['screen_name']):
        tweets_groups.append(list(group))

    tweets_metrics = []
    for group in tweets_groups:
        statuses_count = len(group)
        if statuses_count > TWEETS_MIN_COUNT:
            symbols_count = 0
            polls_count = 0
            hashtags_count = 0
            retweets_count = 0
            media_count = 0
            urls_count = 0
            mentions_count = 0
            sum_text_length = 0
            tweets_count = 0

            for tweet in group:
                hashtags_count += tweet['hashtags']
                media_count += tweet['media']
                urls_count += tweet['urls']
                mentions_count += tweet['user_mentions']
                polls_count += tweet['polls']
                symbols_count += tweet['symbols']
                sum_text_length += tweet['text_length']
                if tweet['is_retweet']:
                    retweets_count += 1
                else:
                    tweets_count += 1
            tweets_metrics.append({
                'avg_hashtags': get_average(hashtags_count, tweets_count),
                'avg_media': get_average(media_count, tweets_count),
                'avg_polls': get_average(polls_count, tweets_count),
                'avg_text_length': get_average(sum_text_length, tweets_count),
                'avg_urls': get_average(urls_count, tweets_count),
                'avg_user_mentions': get_average(mentions_count, tweets_count),
                'retweets_percent': retweets_count / statuses_count,
                'screen_name': group[0]['user'],

            })

    return tweets_metrics
```

Рисунок 2.22 – Функція агрегації трансформованих параметрів повідомлень

```
def merge_metrics(users, tweets):
    results = []
    merged = pandas.merge(
        pandas.DataFrame(users), pandas.DataFrame(tweets), on='screen_name'
    ).to_dict('records')
    for user in merged:
        results.append(omit(user, 'screen_name'))
    return results
```

Рисунок 2.23 – Функція злиття метрик облікового запису та повідомлень

Нижче наведено обраний набір метрик тренувальної вибірки та їх опис (усі метрики мають числовий тип):

- account_age_days – вік облікового запису у днях;
- bot – чи є обліковий запис ботом;
- default_profile – чи змінювався фоновий малюнок профілю;
- default_profile_image – чи змінювалося фото профілю;
- description_length – довжина опису профілю;
- description_url_present – чи присутнє посилання в описі профілю;
- favourites_count – кількість вподовань;
- followers_count – кількість послідовників;
- friends_count – кількість друзів;
- listed_count – кількість публічних списків, куди входить користувач;
- location_present – чи присутня геолокація в профілі;
- name_length – довжина імені;
- protected – чи є обліковий запис захищеним;
- screen_name_length – довжина відображуваного імені;
- statuses_count – кількість повідомлень;
- tweets_per_day – середня кількість повідомлень за день;
- url_present – чи присутнє посилання в профілі;
- verified – чи є обліковий запис верифікованим;
- avg_hashtags – середня кількість хештегів в повідомленнях;
- avg_media – середня кількість медіа контенту в повідомленнях;
- avg_polls – середня кількість опитувань в повідомленнях;
- avg_text_length – середня довжина тексту повідомлення;
- avg_urls – середня кількість посилань в тексті повідомлення;
- avg_user_mentions – середня кількість згадувань інших користувачів;
- retweets_percent – частка перепоширень серед усіх повідомлень.

2.4 Вибір алгоритму навчання моделі

Для тестування знайдених метрик навчальної вибірки було використано наступні алгоритми машинного навчання з бібліотеки scikit-learn для мови програмування Python (рисунок 2.24):

1) Адаптивне підсилення.
2) Дерево ухвалення рішень.
3) Градієнтне підсилення.
4) Метод k-найближчих сусідів (англ. k-nearest neighbor method)
5) Багатошаровий перцептрон Румельхарта.
6) Випадковий ліс.

```
from sklearn.ensemble import (
    AdaBoostClassifier,
    RandomForestClassifier,
    GradientBoostingClassifier,
)
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier


MODELS = {
    'Adaptive boosting': AdaBoostClassifier(n_estimators=100),
    'Decision tree': DecisionTreeClassifier(),
    'Gradient boosting': GradientBoostingClassifier(),
    'KNN': KNeighborsClassifier(n_neighbors=5),
    'MLP': MLPClassifier(),
    'Random forest': RandomForestClassifier(n_estimators=100),
}
```

Рисунок 2.24 – Ініціалізація обраних алгоритмів навчання

Якість обраних метрик та, відповідно, точність побудованої моделі ідентифікації ботів перевірялася спочатку на кожному з перелічених алгоритмів машинного навчання з використанням 10-кратної перехресної валідації.

Для кожного з алгоритмів за результатами валідації було обраховано два показники (їх формальне визначення наведено в розділі 3 даної роботи):

- правильність передбачення класу облікового запису для тестової вибірки (accuracy);
- стандартне відхилення отриманих передбачень (standard deviation).

Однією з особливостей побудованої моделі є автоматичний вибір алгоритму навчання, який показав найкращий показник точності передбачень при перехресній валідації. На рисунку 2.25 зображено програмний код вибору найкращого алгоритму навчання.

```
def select_best_algo(x_train, y_train):
    print('Trying to determine best algorithm with 10-fold cross-validation.')
    print(LINE_DELIMITER)
    results = []
    scoring = 'accuracy'
    seed = random.randint(1, 100)
    for model_type, model in MODELS.items():
        print(f'Test {model_type}')
        kfold = KFold(n_splits=10, random_state=seed)
        cv_results = cross_val_score(
            model, x_train, y_train, cv=kfold, scoring=scoring
        )
        accuracy = cv_results.mean()
        standard_deviation = cv_results.std()
        print(f'Accuracy: {accuracy}')
        print(f'Standard deviation: {standard_deviation}')
        print(LINE_DELIMITER)
        results.append({
            'algorithm_name': model_type,
            'accuracy': accuracy,
            'standard deviation': standard_deviation,
        })
    return results
```

Рисунок 2.25 – Функція перевірки алгоритму навчання методом перехресної валідації

Наступним кроком є власне навчання за обраним на попередньому кроці алгоритмом на виділеній тренувальній вибірці та подальша перевірка точності обчислень на тестовій вибірці.

Поширеним рішенням при використанні машинного навчання є розбиття навчальної вибірки на дві частки: набір, на якому ведеться навчання, та набір, на якому тестується якість проведеного тренування. При перевірці точності обраних метрик в даній роботі використовувався такий же підхід. Перед розбиттям навчальна вибірка було попередньо оброблена та підготовлена (рисунок 2.26).

```
def make_df(data):
    df = pandas.DataFrame(data)
    columns = df.columns.tolist()
    columns.append(columns.pop(columns.index('bot')))
    return df[columns]


def prepare_data(df, validation_size):
    seed = random.randint(1, 100)
    array = df.values
    features = array[:, 0:-1]
    predict = array[:, -1].astype('int')
    return train_test_split(
        features, predict, test_size=validation_size, random_state=seed
    )
```

Рисунок 2.26 – Підготовка та розбиття навчальної вибірки на тренувальну та тестову складові

Вхідними параметрами відповідного програмного забезпечення є шлях до файлу, що містить тренувальну вибірку та частка цього набору, що буде використовуватися в якості тестової вибірки.

Вихідними оцінками якості є правильність, точність (англ. precision), повнота (англ. recall) та F-міра (англ. f1-score) для передбачень кожного класу. Формальне визначення даних показників наведено в розділі 3 даної роботи. Програмний код різних складових тестування якості побудованої моделі наведено на рисунку 2.27.

```
@click.command()
@click.option('--filename', help='Path to file with training data.')
@click.option(
    '--validation_size',
    help='Percentage of data that will be used for validation.',
    type=float
)
def test_metrics(filename, validation_size):
    data = get_data(filename)
    df = make_df(data)
    x_train, x_test, y_train, y_test = prepare_data(df, validation_size)
    algorithm_name = first(sorted(
        select_best_algo(x_train, y_train),
        key=lambda x: x['accuracy'],
        reverse=True
    ))['algorithm_name']
    model = MODELS[algorithm_name]
    print(f'{algorithm_name} was selected.')
    model.fit(x_train, y_train)
    predictions = model.predict(x_test)
    print(f'Accuracy of predictions: {accuracy_score(y_test, predictions)}')
    print(classification_report(y_test, predictions, digits=4))
```

Рисунок 2.27 – Команда тестування якості метрик

Висновки до розділу 2

В даному розділі було детально розглянуто існуючі тренувальні набори для розв’язання задачі класифікації облікових записів Twitter. В ході проведеного аналізу виявлено, що більшість вибірок є застарілої та неповною. Було сформовано власну навчальну вибірку.

За результатами проведених досліджень залежності параметрів облікового запису та класу, до якого він належить, було запропоновано множину метрик, які найкраще показують різницю між ботом та справжнім користувачем.

В ході подальшої роботи було створено програмне забезпечення для тестування відібраних метрик за заданими алгоритмами машинного навчання.

3 АНАЛІЗ ЯКОСТІ ПОБУДОВАНОЇ МОДЕЛІ

3.1 Показники якості класифікації

Якість отриманих метрик, що моделюють поведінку бота в соціальній мережі Twitter, було оцінено за трьома основними параметрами: повнота (або вичерпність), точність та їх гармонічне середнє. В теорії машинного навчання це відповідно метрики recall, precision та f1-score (f-score, f-measure), що вже були згадані в попередньому розділі. Для вибору оптимального алгоритму навчання використовувалася метрика правильності.

Метрика правильності класифікатора – це співвідношення правильно визначених об’єктів (без прив’язки до класу) до загальної кількості об’єктів. Формальне визначення:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

де TP – кількість істинно-позитивних рішень;

TN – кількість істинно-негативних рішень;

FP – кількість хибно-позитивних рішень;

FN – кількість хибно-негативних рішень.

Метрика точності моделі у межах оцінюваного класу це співвідношення об’єктів, що дійсно належать класу, до всіх об’єктів, що були віднесені системою до класу. Формальне описання:

$$Precision = \frac{TP}{TP+FP} \tag{3.2}$$

Метрика повноти моделі у межах класу це співвідношення знайдених класифікатором об’єктів, що належать класу, до всіх об’єктів, що належать цьому класу. Формальне описання:

$$Recall = \frac{TP}{TP+FN} \tag{3.3}$$

F-міра – це метрика, що об’єднує показники повноти та точності моделі та являє собою гармонічне середнє цих параметрів:

$$F = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{3.4}$$

Деякі з описаних в розділі 1 моделей ідентифікації оперують також параметром AUC, тому для порівняння якості розробленої моделі його також було обраховано.

Метрика AUC – це площа під ROC-кривою – графіком, який дозволяє оцінити якість бінарної класифікації та являє собою співвідношення показника повноти моделі до показника випадання (англ. fall-out або false positive rate). Показник випадання моделі – це ймовірність того, що об’єкт, який не належить до класу, буде туди віднесено. Чим вище показник AUC – тим краща точність класифікації.

3.2 Порівняння з існуючими моделями

Результати етапу відбору кращого алгоритму, з точністю до 4 знаків після коми, подано у таблиці 3.1.

Таблиця 3.1 – Результати 10-кратної перехресної перевірки.

Назва алгоритму	**Accuracy**	**Standard deviation**
Адаптивне підсилення	0.9417	0.0065
Дерево ухвалення рішень	0.9203	0.0065
Градієнтне підсилення	0.9464	0.0063
Метод k-найближчих сусідів	0.8504	0.0079
Багатошаровий перцептрон Румельхарта	0.8408	0.0262
Випадковий ліс	0.9475	0.0040

Для подальшого оцінювання моделі було обрано алгоритм випадкового лісу.

Вихідні оцінки якості моделі для частки тестової вибірки 10% становили (з точністю до 4 знаків після коми):

- accuracy: 0.9398;
- метрики для класу «бот» (розмір тестової вибірки – 542 облікових записи):
  - precision: 0.9242;
  - recall: 0.8321;
  - f1-score: 0.8757;
- метрики для класу «людина» (розмір тестової вибірки – 1586 облікових записів):
  - precision: 0.9445;
  - recall: 0.9767;
  - f1-score: 0.9603;

- AUC: 0.9044;
- кількість істинно-негативних рішень: 1549;
- кількість істинно-позитивних рішень: 451;
- кількість хибно-негативних рішень: 91;
- кількість хибно-позитивних рішень: 37.

Проведемо порівняльний аналіз побудованої моделі з існуючими, що були представлені в розділі 1 даної роботи.

1) Онлайн сервіс Botcheck.me.

Автори даної моделі наводять лише два параметри точності своєї розробки: accuracy та первинний показник похибки першого роду (на самому початку запуску сервісу). Також автори не повідомляють обраний алгоритм навчання. В таблиці 3.2 представлено порівняння наявних показників точності botcheck.me з побудованою моделлю:

Таблиця 3.2 – Порівняння побудованої моделі з сервісом Botcheck.me

Показник точності	**Побудована модель**	**Botcheck.me**
accuracy	0.9398	0.935
false positive error	0.0174	0.02

З наведених результатів видно, що побудована модель показує кращі показники точності. Це може пояснюватися двома причинами: 1) як зазначалося в розділі 1, автори botcheck.me згенерували частину навчальної вибірки ботів за допомогою евристичних правил серед послідовників інших ботів, які були помічені вручну, - це могло призвести до погіршення якості вибірки і, як наслідок, загальної точності моделі; 2) також, за словами авторів, в моделі botcheck.me було використано більше 100 параметрів для класифікації – такий надлишок міг призвести не до покращення точності, а навпаки, до зменшення впливу тих метрик, які дійсно маюсь значення.

2) Онлайн сервіс Botometer.

Автори наводять значно більше показників точності, їх порівняльний аналіз з побудованою моделлю наведено у таблиці 3.3:

Таблиця 3.3 – Порівняння побудованої моделі з сервісом Botometer.

Показник	**Побудована модель**	**Botometer**
AUC	0.9044	0.89
false positive rate	0.0233	0.15
false negative rate	0.1677	0.11
Алгоритм навчання	Випадковий ліс	Випадковий ліс

Наведені результати показують, що побудована модель має значно кращий результат у виявленні людей: показник false positive rate – це ймовірність того, що обліковий запис реальної людини буде ідентифіковано як бота. В той же час, побудована модель показує себе гірше у виявленні ботів: ймовірність того, що бота буде ідентифіковано як реальну особу становить 0.1677. Але в цілому, якщо брати до уваги обидва показники, що об’єднуються показником AUC, то можна побачити, що побудована модель знову показує кращу точність розрізнення ботів та людей.

3) Бібліотека TweetBotOrNot.

Автор моделі наводить показники precision (для обох класів) та accuracy для обох варіацій своєї моделі. Порівняння оцінок точності моделей наведено в таблиці 3.4:

Таблиця 3.4 – Порівняння побудованої моделі з бібліотекою TweetBotOrNot

Модель	**Precision (боти)**	**Precision (люди)**	**Accuracy**
Швидка варіація TweetBotOrNot	0.9178	0.9261	0.919
Стандартна варіація TweetBotOrNot	0.9353	0.9532	0.938
Побудована модель	0.9242	0.9445	0.9398

Не зважаючи на дещо кращі показники точності для стандартної модифікації, загальний показник правильності визначення більший у побудованої моделі. Оскільки автор не наводить інших оцінок точності, можна лише припустити, що загальна оцінка правильності нижча через недостатню повноту визначення. В обох варіаціях моделі TweetBotOrNot автор використав метод градієнтного підсилення в якості алгоритму навчання.

Під час порівняння моделей, через брак відомостей про існуючі сервіси та бібліотеки, не всі вихідні оцінки якості запропонованої моделі були інтерпретовані. Нижче наведено пояснення щодо параметрів якості та точності, що не були розглянуті:

- Метрика recall. Отримані значення свідчать радше про недостатній обсяг частки ботів у навчальній вибірці, аніж про неправильно обрані метрики. Для реальних користувачів, при їх кількості в три рази вищій, даний параметр становив 0.9767, що є показником достатньої повноти знань, а отже й вичерпності обраних метрик.
- F-міра. Значення даного показника є балансом точності та повноти моделі. Отримані значення для ботів свідчать про те, що підвищення загальної правильності визначення об’єктів цього класу можна досягти в

першу чергу за рахунок підвищення показника повноти моделі для автоматизованих облікових записів, а отже підвищення частки ботів в навчальній вибірці.

Результати порівняльного аналізу моделей свідчать про вищу якість та точність розробленого рішення у порівнянні з існуючими. Серед особливостей побудованої моделі, які було вперше отримано у даній роботі та які дозволили покращити існуючі показники точності, правильності та якості ідентифікації ботів, можна виділити:

- Вибір класифікуючих метрик облікового запису, яким в попередніх дослідженнях не було приділено уваги, а саме: довжини імені, опису профілю, відображуваного імені; середня кількість опитувань та медіа контенту в повідомленнях; факт наявності посилань в описі облікового запису.
- Новий підхід до процесу вибору алгоритму машинного навчання, який дозволяє легко розширювати модель за рахунок нових алгоритмів та завжди обирати той метод навчання, який показує найвищі показники точності класифікації.
- Актуалізація навчальної вибірки за допомогою розробленого програмного забезпечення.
- Врахування параметрів облікових записів публічних людей, які, як показав проведений аналіз, мають свої особливості у порівнянні зі звичайними користувачами та ботами.

‘

3.3 Тестування моделі в реальних умовах

Всі наведені показники якості та точності побудованої моделі ідентифікації ботів були отримані на тестовій вибірці, що є часткою навчальної вибірки, але частиною будь-якої перевірки моделі, що використовує засоби машинного навчання, є апробація результатів на даних в «диких умовах» (англ. in the wild).

Для такого тестування було вручну відібрано 42 облікових записи ботів та реальних користувачів: по 21 на кожний клас. Такий набір представив тестову вибірку, при цьому в якості навчальної вибірки було використано той же самий набір, але з урахуванням тої частки, що раніше була обрана в якості тестового набору.

Результати 10-кратної перехресної перевірки на розширеній навчальній вибірці наведено у таблиці 3.5:

Таблиця 3.5 – Результати 10-кратної перехресної перевірки на розширеній навчальній вибірці.

Назва алгоритму	**Accuracy**	**Standard deviation**
Адаптивне підсилення	0.9222	0.0721
Дерево ухвалення рішень	0.9010	0.0592
Градієнтне підсилення	0.9271	0.0761
Метод k-найближчих сусідів	0.8049	0.1777
Багатошаровий перцептрон Румельхарта	0.8071	0.1633
Випадковий ліс	0.9253	0.0813

Як видно з таблиці, найкращий показник точності було отримано за допомогою алгоритму градієнтного підсилення.

Показники точності при визначенні класу обраних 42 облікових записів набули наступних значень:

- клас «бот»:
  - precision: 1.0000;
  - recall: 0.9524;
  - f1-score: 0.9767;
- клас «людина»:
  - precision: 0.9545;
  - recall: 1.0000;
  - f1-score: 0.9756;
- accuracy: 0.9762;
- кількість true negative: 21;
- кількість false positive: 0;
- кількість false negative: 1;
- кількість true positive: 20.
- AUC: 0.9762.

Результати показали, що з-поміж 42 випадково обраних облікових записів модель правильно визначила всіх реальних користувачів та допустила лише 1 помилку у визначенні ботів – бота з іменем RedScareBot було класифіковано як реального користувача.

Такі результати лише підтверджують якісність та точність побудованої моделі.

Висновки до розділу 3

Отримані результати порівняльного аналізу побудованої моделі з існуючими рішеннями свідчать про високу конкурентну спроможність запропонованого підходу до процесу ідентифікації ботів.

В ході дослідження якості обраних метрик для вирішення задачі класифікації автоматизованих облікових записів було встановлено, що вони цілком відповідають вимогам повноти та точності.

Одним із способів покращення точності розробленої моделі може бути збалансування навчальної вибірки без втрати її загального обсягу.

ВИСНОВКИ

В даній роботі було досліджено моделі розпізнавання ботів у соціальній мережі Twitter.

В ході роботи було проведено аналіз структури та принципу роботи існуючих рішень з ідентифікації ботів. Було з’ясовано, що показники точності та якості розглянутих моделей мають ряд недоліків.

В результаті проведеної роботи було розроблено програмне забезпечення для отримання необхідних для ідентифікації даних, визначено параметри, які характеризують автоматизований обліковий запис в соціальній мережі Twitter, побудовано модель розпізнавання ботів.

Наукові результати отримані в роботі підтверджують ряд гіпотез, що присутні в дослідженнях авторів існуючих моделей ідентифікації, стосовно моделі бота в соціальних мережах.

Поставлені мету та завдання на роботу виконано у повному обсязі: було побудовано модель ідентифікації ботів у соціальній мережі Twitter з удосконаленими показниками точності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Onur Varol. Online Human-Bot Interactions: Detection, Estimation, and Characterization [Text] / Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, Alessandro Flammini // Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017), Montreal, Canada, May 15–18, 2017, – P. 280 – 290.

2. RoBhat Labs. Identifying Propaganda Bots on Twitter [Electronic resource] / RoBhat Labs – Access mode: https://medium.com/@robhat

3. RoBhat Labs. An Analysis of Propaganda Bots on Twitter [Electronic resource] / RoBhat Labs – Access mode: https://medium.com/@robhat

4. Kyumin Lee. Seven Months with the Devil: A Long-Term Study of Content Polluters on Twitter [Text] / Kyumin Lee, Brian David Eoff, James Caverlee // Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 2011), Barcelona, Spain, July 17–21, 2011.

5. Michael W. Kearney. TweetBotOrNot [Electronic resource] / Michael W. Kearney – Access mode: https://mikewk.shinyapps.io/botornot/

6. Michael W. Kearney. TweetBotOrNot GitHub repository [Electronic resource] / Michael W. Kearney – Access mode: https://github.com/mkearney/tweetbotornot

7. NYU Tandon Spring 2017 Machine Learning Competition: Twitter Bot classification [Electronic resource] / Access mode: https://www.kaggle.com/c/twitter-bot-classification

8. Charvi Jain. Detecting twitter bot data [Electronic resource] / Charvi Jain – Access mode: https://www.kaggle.com/charvijain27

9. S. Cresci. Fame for sale: efficient detection of fake Twitter followers [Text] / S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi // Elsevier Decision Support Systems, – Vol. 80, – December 2015 – P. 56–71.

10. S. Cresci. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race [Text] / S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi //

WWW '17 Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 03 – 07, 2017 – P. 963-972.

11. C Yang. Analyzing Spammer's Social Networks for Fun and Profit: A Case Study of Cyber Criminal Ecosystem on Twitter [Text] / C Yang, R Harkreader, J Zhang, S Shin, G Gu // WWW '12 Proceedings of the 21st international conference on World Wide Web, Lyon, France, April 16 – 20, 2012 – P. 71-80.

12. Gilani Zafar. Of Bots and Humans (on Twitter) [Text] / Gilani Zafar. Farahbakhsh Reza, Tyson Gareth, Wang Liang, Crowcroft Jon // Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, Sydney, Australia, July 31 – August 03, 2017 – P. 349 – 354.

13. Gilani Zafar. Classification of Twitter Accounts into Automated Agents and Human Users [Text] / Gilani Zafar, Kochmar Ekaterina, Crowcroft Jon // Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, Sydney, Australia, July 31 – August 03, 2017 – P. 489 – 496.

14. Jason Baumgartner. Jason Baumgartner's Twitter page [Electronic resource] / Jason Baumgartner – Access mode: https://twitter.com/jasonbaumgartne